

# SIEMENS

## SIMATIC

### S7 S7-1200 Automatisierungssystem


#### Systemhandbuch


Vorwort	
Produktübersicht	1
STEP 7 Programmiersoftware	2
Einbau	3
SPS-Grundlagen	4
Gerätekonfiguration	5
Programmiergrundlagen	6
Anweisungen	7
Erweiterte Anweisungen	8
Technologieanweisungen	9
Kommunikation	10
Webserver	11
Kommunikationsprozessor	12
Teleservice-Kommunikation (SMTP-E-Mail)	13
Online- und Diagnose-Tools	14
Technische Daten	A
Berechnen der Leistungsbilanz	B
Bestellnummern	C


## Rechtliche Hinweise

### Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 <b>GEFAHR</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>wird</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>WARNUNG</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>kann</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>VORSICHT</b>
mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

<b>VORSICHT</b>
ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

<b>ACHTUNG</b>
bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.


Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

### Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

### Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 <b>WARNUNG</b>
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

### Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

### Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

# Vorwort

## Zweck des Handbuchs

Die Familie S7-1200 umfasst verschiedene speicherprogrammierbare Steuerungen (SPS), mit denen Sie eine breite Palette von Automatisierungsaufgaben lösen können. Durch das kompakte Design, die Möglichkeit der Erweiterung, den günstigen Preis und einen leistungsstarken Befehlssatz eignet sich die S71200 hervorragend für eine Vielzahl von Steuerungsanwendungen. Die Ausführungen der S7-1200 und die vielfältigen Programmiermöglichkeiten unter Windows bieten Ihnen extrem hohe Flexibilität beim Umsetzen Ihrer Automatisierungslösungen.

Dieses Handbuch bietet Informationen zum Installieren und Programmieren von S7-1200 Steuerungen und wendet sich an Ingenieure, Programmierer und Wartungspersonal mit allgemeinen Kenntnissen über Automatisierungssysteme.

## Erforderliche Grundkenntnisse

Damit Sie mit diesem Handbuch arbeiten können, benötigen Sie allgemeine Kenntnisse im Bereich der Automatisierung und der speicherprogrammierbaren Steuerungen.

## Umfang des Handbuchs

Dieses Handbuch beschreibt die folgenden Produkte:

- STEP 7 V11 Basic und Professional
- S7-1200 CPU Firmware-Version V3.0

Eine vollständige Liste der S71200 Produkte finden Sie in den technischen Daten (Seite 743).

## Zertifizierung, CE-Kennzeichen, C-Tick und andere Normen

Ausführliche Informationen finden Sie in den technischen Daten (Seite 743).

## Service & Support

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser technisches Know-how auf der Kundensupport-Website (<http://www.siemens.com/automation/>) an.

Falls Sie technische Fragen haben, eine Schulung benötigen oder S7-Produkte bestellen wollen, wenden Sie sich bitte an Ihre Siemens-Vertretung. Das technisch geschulte Vertriebspersonal verfügt über sehr spezifische Kenntnisse zu Einsatzmöglichkeiten und Prozessen sowie zu den verschiedenen Siemens-Produkten und kann Ihnen deshalb am schnellsten und besten weiterhelfen, wenn Probleme auftreten.

## Dokumentation und Information

S7-1200 und STEP 7 bieten eine Vielzahl von Dokumentationen und anderen Quellen mit technischen Informationen.

- Das S7-1200 Systemhandbuch bietet spezifische Informationen zu Funktionsweise, Programmierung und technischen Daten der gesamten S7-1200 Produktreihe. Neben dem Systemhandbuch bietet das S7-1200 Easy Book einen eher allgemeinen Überblick über die Fähigkeiten der S7-1200 Familie.

Sowohl das Systemhandbuch als auch das Easy Book stehen elektronisch (im PDF-Format) und als gedrucktes Handbuch zur Verfügung. Die elektronischen Handbücher können von der Kundensupport-Website heruntergeladen werden. Sie finden sie außerdem auf den Datenträgern, die mit jeder S7-1200 CPU ausgeliefert werden.

- Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Konzeptinformationen und spezifische Anweisungen zu Funktionsweise und Funktionalität des Programmierpakets sowie zur grundlegenden Funktionsweise der SIMATIC CPUs.
- Mit My Documentation Manager greifen Sie auf die elektronische Version (PDF) des SIMATIC Dokumentationssatzes zu, einschließlich Systemhandbuch, Easy Book und Informationssystem von STEP 7. In My Documentation Manager können Sie Themen aus verschiedenen Dokumenten per Drag&Drop anordnen und so eigene benutzerspezifische Handbücher anlegen.

Im Kundensupport-Portal (<http://support.automation.siemens.com>) finden Sie einen Link auf My Documentation Manager unter mySupport.

- Die Kundensupport-Website bietet außerdem Podcasts, FAQs und andere hilfreiche Dokumente für S7-1200 und STEP 7. Die Podcasts zeigen kurze Lernvideos zu spezifischen Funktionen oder Szenarien, um die Interaktionen, die praktische Bedienung und Effizienz von STEP 7 zu demonstrieren. Die Sammlung der Podcasts finden Sie auf folgenden Websites:
  - Webseite zu STEP 7 Basic (<http://www.automation.siemens.com/mcms/simatic-controller-software/de/step7/step7-basic/Seiten/Default.aspx>)
  - Webseite zu STEP 7 Professional (<http://www.automation.siemens.com/mcms/simatic-controller-software/de/step7/step7-professional/Seiten/Default.aspx>)
- Im technischen Forum unter Service & Support (<https://www.automation.siemens.com/WW/forum/guests/Conferences.aspx?Language=de&onlyInternet=False>) können Sie außerdem Produktdiskussionen verfolgen oder sich daran beteiligen. Folgende Foren bieten Ihnen die Möglichkeit, mit verschiedenen Produktexperten in Kontakt zu treten.
  - Forum für S7-1200 (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=257&Language=de&onlyInternet=False>)
  - Forum für STEP 7 Basic (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=262&Language=de&onlyInternet=False>)

# Inhaltsverzeichnis

	<b>Vorwort</b> .....	<b>3</b>
<b>1</b>	<b>Produktübersicht</b> .....	<b>19</b>
1.1	Einführung in die S7-1200 SPS .....	19
1.2	Erweiterung der CPU-Funktionen .....	22
1.3	S7-1200 Module.....	25
1.4	Neue Funktionen.....	27
1.5	Grundlegende HMI-Panels .....	28
<b>2</b>	<b>STEP 7 Programmiersoftware</b> .....	<b>31</b>
2.1	Systemvoraussetzungen.....	32
2.2	Einfaches Arbeiten mit unterschiedlichen Ansichten .....	32
2.3	Bedienerfreundliche Werkzeuge.....	33
2.3.1	Anweisungen in Ihr Anwenderprogramm einfügen.....	33
2.3.2	Zugreifen auf Anweisungen in der Funktionsleiste "Favoriten" .....	34
2.3.3	Erstellen einer komplexen Gleichung mit einer einfachen Anweisung .....	35
2.3.4	Ergänzen von Eingängen oder Ausgängen in einer KOP- oder FUP-Anweisung .....	37
2.3.5	Erweiterbare Anweisungen .....	37
2.3.6	Auswählen einer Version für eine Anweisung .....	38
2.3.7	Ändern des Erscheinungsbilds und der Konfiguration von STEP 7 .....	39
2.3.8	Drag & Drop zwischen Editoren.....	39
2.3.9	Wechseln des Betriebszustands der CPU .....	40
2.3.10	Aufruftyp eines DB ändern.....	41
2.3.11	Geräte vorübergehend vom Netzwerk trennen.....	42
2.3.12	Virtuelles Abziehen von Geräten aus der Konfiguration.....	43
<b>3</b>	<b>Einbau</b> .....	<b>45</b>
3.1	Richtlinien für den Einbau von S71200 Geräten.....	45
3.2	Leistungsbilanz .....	47
3.3	Ein- und Ausbauanweisungen .....	49
3.3.1	Einbaumaße der S7-1200 Geräte.....	49
3.3.2	Einbau und Ausbau der CPU.....	52
3.3.3	Ein- und Ausbau eines SBs, CBs oder BBs .....	55
3.3.4	Einbau und Ausbau eines SMs.....	57
3.3.5	Einbau und Ausbau eines CMs oder CPs .....	59
3.3.6	Ausbau und Wiedereinbau des S7-1200 Klemmenblocks .....	60
3.3.7	Einbau und Ausbau des Erweiterungskabels .....	61
3.3.8	TS (Teleservice)-Adapter .....	63
3.3.8.1	Anschließen des TeleService-Adapters.....	63
3.3.8.2	Einlegen der SIM-Karte.....	64
3.3.8.3	Einbau des TS-Adapters.....	66
3.3.8.4	Wandmontage des TS-Adapters.....	66

3.4	Verdrahtungsrichtlinien .....	67
<b>4</b>	<b>SPS-Grundlagen.....</b>	<b>73</b>
4.1	Ausführung des Anwenderprogramms .....	73
4.1.1	Betriebszustände der CPU.....	75
4.1.2	Bearbeitung des Zyklus im Betriebszustand RUN .....	79
4.1.3	Organisationsbausteine (OBs).....	80
4.1.4	Prioritäten und Warteschlange für die Ausführung von Ereignissen .....	81
4.1.5	Überwachen der Zykluszeit.....	87
4.1.6	CPU-Speicher .....	89
4.1.6.1	System- und Taktmerker.....	92
4.1.7	Diagnosepuffer.....	94
4.1.8	Echtzeituhr .....	94
4.1.9	Konfigurieren der Ausgänge für den Wechsel von RUN in STOP.....	95
4.2	Datenspeicher, Speicherbereiche, E/A und Adressierung.....	95
4.2.1	Zugriff auf Daten der S7-1200 .....	95
4.3	Verarbeitung von Analogwerten.....	101
4.4	Datentypen.....	102
4.4.1	Datentypen Bool, Byte, Word und DWord .....	103
4.4.2	Ganzzahlige Datentypen.....	104
4.4.3	Gleitpunktzahl/Realzahl-Datentypen.....	104
4.4.4	Uhrzeit- und Datums-Datentypen .....	105
4.4.5	Zeichen- und Zeichenfolge-Datentypen.....	106
4.4.6	Datentyp ARRAY .....	108
4.4.7	Datentyp Struktur .....	109
4.4.8	PLC-Datentyp.....	109
4.4.9	Pointer-Datentypen .....	110
4.4.9.1	Pointer-Datentyp "Pointer" .....	110
4.4.9.2	Pointer-Datentyp "Any" .....	111
4.4.9.3	Pointer-Datentyp "Variant" .....	113
4.4.10	Zugriff auf eine "Scheibe" eines Variablentyps .....	113
4.4.11	Zugriff auf eine Variable mit einer AT-Überlagerung .....	115
4.5	Memory Card verwenden.....	116
4.5.1	Memory Card in die CPU stecken.....	117
4.5.2	Anlaufparameter der CPU vor dem Kopieren des Projekts auf die Memory Card konfigurieren .....	119
4.5.3	Einsatz der Memory Card als Übertragungskarte.....	119
4.5.4	Einsatz der Memory Card als Programmkarte.....	121
4.5.5	Firmware-Update .....	124
4.6	Vorgehensweise bei verlorenem Passwort.....	127
<b>5</b>	<b>Gerätekonfiguration .....</b>	<b>129</b>
5.1	CPU einfügen.....	130
5.2	Konfiguration für eine nicht spezifizierte CPU erkennen .....	131
5.3	Module zur Konfiguration hinzufügen .....	132
5.4	Konfigurieren des CPU-Betriebs.....	133
5.5	Modulparameter konfigurieren .....	136

5.6	CPU für die Kommunikation konfigurieren.....	137
5.6.1	Netzwerkverbindung erstellen.....	137
5.6.2	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren.....	138
5.6.3	Parameter für die PROFINET-Verbindung .....	140
5.6.4	IP-Adressen zuweisen .....	143
5.6.4.1	IP-Adressen zu Programmier- und Netzwerkgeräten zuweisen.....	143
5.6.4.2	Ermitteln der IP-Adresse Ihres Programmiergeräts.....	145
5.6.4.3	Online eine IP-Adresse zu einer CPU zuweisen .....	145
5.6.4.4	IP-Adresse für eine CPU in Ihrem Projekt konfigurieren .....	147
5.6.5	Testen des PROFINET-Netzwerks .....	150
5.6.6	Ermitteln der Ethernet-Adresse (MAC-Adresse) der CPU.....	151
5.6.7	Konfigurieren der NTP-Synchronisierung .....	152
5.6.8	Anlaufzeit, Benennung und Adresszuweisung von PROFINET-Geräten.....	153
<b>6</b>	<b>Programmiergrundlagen .....</b>	<b>155</b>
6.1	Richtlinien für das Entwerfen einer Automatisierungslösung mit einem PLC-Gerät .....	155
6.2	Strukturieren Ihres Anwenderprogramms .....	156
6.3	Strukturierung des Programms mit Hilfe von Bausteinen.....	158
6.3.1	Organisationsbaustein (OB).....	159
6.3.2	Funktion (FC) .....	160
6.3.3	Funktionsbaustein (FB).....	161
6.3.4	Datenbaustein (DB) .....	162
6.4	Datenkonsistenz .....	163
6.5	Programmiersprache .....	165
6.5.1	Kontaktplan (KOP) .....	165
6.5.2	Funktionsplan (FUP) .....	166
6.5.3	SCL .....	166
6.5.4	EN und ENO in KOP, FUP und SCL .....	174
6.6	Schutz .....	176
6.6.1	Zugriffsschutz für die CPU .....	176
6.6.2	Knowhow-Schutz .....	177
6.6.3	Kopierschutz .....	178
6.7	Laden der Programmelemente .....	179
6.8	Laden aus der CPU .....	179
6.8.1	Elemente des Projekts kopieren .....	179
6.8.2	Mit der Vergleichsfunktion arbeiten .....	181
6.9	Debugging und Testen des Programms .....	181
6.9.1	Daten in der CPU beobachten und steuern.....	181
6.9.2	Beobachtungstabellen und Forcetabellen .....	182
6.9.3	Querverweis zum Anzeigen der Verwendung .....	182
6.9.4	Aufrufstruktur zur Prüfung der Aufrufhierarchie.....	183
<b>7</b>	<b>Anweisungen .....</b>	<b>185</b>
7.1	Bitverknüpfung .....	185
7.1.1	Bitverknüpfungskontakte und -spulen.....	185
7.1.2	Setz- und Rücksetzoperationen.....	188
7.1.3	Operationen Steigende Flanke und Fallende Flanke .....	191
7.2	Zeiten .....	193

7.3	Zähler .....	201
7.4	Vergleichen .....	207
7.4.1	Vergleichen .....	207
7.4.2	Operationen IN_RANGE und OUT_RANGE .....	208
7.4.3	Operationen OK und NOT_OK .....	208
7.5	Arithmetik .....	209
7.5.1	Anweisung Berechnen .....	209
7.5.2	Operationen Addieren, Subtrahieren, Multiplizieren und Dividieren .....	211
7.5.3	Modulo-Anweisung .....	212
7.5.4	Anweisung Negation .....	213
7.5.5	Operationen INC (Inkrementieren) und DEC (Dekrementieren) .....	213
7.5.6	Operation ABS (Absolutwert bilden) .....	214
7.5.7	Anweisungen MIN und MAX .....	215
7.5.8	Anweisung LIMIT .....	216
7.5.9	Arithmetische Gleitpunktoperationen .....	216
7.6	MOVE .....	218
7.6.1	Operationen MOV (Übertragen) und MOVE_BLK (Wertebereich übertragen) .....	218
7.6.2	Anweisungen FeldLesen und FeldSchreiben .....	221
7.6.3	Operationen Speicher mit Bitmuster belegen .....	223
7.6.4	Operation SWAP (Bytes tauschen) .....	224
7.7	Umwandeln .....	225
7.7.1	Anweisung CONV .....	225
7.7.2	Umwandlungsanweisungen in SCL .....	226
7.7.3	Operationen ROUND (Zahl runden) und TRUNC (Ganzzahl erzeugen) .....	229
7.7.4	Operationen CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen) und FLOOR (Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen) .....	230
7.7.5	Operationen Skalieren und Normalisieren .....	231
7.8	Programmsteuerung .....	234
7.8.1	Übersicht über die Programmsteuerungsanweisungen in SCL .....	234
7.8.2	IF-THEN-Anweisung .....	235
7.8.3	CASE-Anweisung .....	236
7.8.4	FOR-Anweisung .....	237
7.8.5	WHILE-DO-Anweisung .....	238
7.8.6	REPEAT-UNTIL-Anweisung .....	239
7.8.7	CONTINUE-Anweisung .....	240
7.8.8	EXIT-Anweisung .....	241
7.8.9	GOTO-Anweisung .....	241
7.8.10	RETURN-Anweisung .....	242
7.8.11	Operationen Sprung und Sprungmarke .....	242
7.8.12	Anweisung JMP_LIST .....	243
7.8.13	Anweisung SWITCH .....	244
7.8.14	Programmsteuerungsoperation RET .....	246
7.8.15	Zykluszeitüberwachung neu starten .....	247
7.8.16	Zyklus anhalten .....	248
7.8.17	Operationen Get_Error .....	248
7.9	Wortverknüpfung .....	252
7.9.1	Operationen AND (UND), OR (ODER) und XOR (EXKLUSIV ODER) .....	252
7.9.2	Operation INV (Einerkomplement erzeugen) .....	253
7.9.3	Operationen Hexadezimalzahl in Bit wandeln und Bit in Hexadezimalzahl wandeln .....	254



7.9.4	Anweisungen Selektieren, Multiplexen und Demultiplexen .....	255
7.10	Schieben und Rotieren .....	258
7.10.1	Schiebeoperationen .....	258
7.10.2	Rotieroperationen .....	259
<b>8</b>	<b>Erweiterte Anweisungen .....</b>	<b>261</b>
8.1	Datum und Uhrzeit .....	261
8.1.1	Datum- und Uhrzeitoperationen .....	261
8.1.2	Systemuhr einstellen und lesen .....	263
8.1.3	Betriebsstundenzähler .....	265
8.1.4	Anweisung SET_TIMEZONE .....	267
8.2	Zeichenketten- und Zeichenanweisungen .....	268
8.2.1	Datentyp String .....	268
8.2.2	Anweisung S_MOVE .....	269
8.2.3	Zeichenkettenkonvertierungsoperationen .....	269
8.2.3.1	Konvertierung Zeichenketten in Werte und Werte in Zeichenketten .....	269
8.2.3.2	Umwandlung Zeichenkette-in-Zeichen und Zeichen-in-Zeichenkette .....	278
8.2.3.3	Umwandlung von ASCII-Zeichenketten in Hexadezimalzahlen und von Hexadezimalzahlen in ASCII-Zeichenketten .....	280
8.2.4	Zeichenkettenanweisungen .....	282
8.2.4.1	LEN .....	282
8.2.4.2	CONCAT .....	283
8.2.4.3	LEFT, RIGHT und MID .....	284
8.2.4.4	DELETE .....	285
8.2.4.5	INSERT .....	286
8.2.4.6	REPLACE .....	287
8.2.4.7	FIND .....	288
8.3	Dezentrale E/A (PROFINET, PROFIBUS oder AS-i) .....	289
8.3.1	Anweisungen für die dezentrale E/A .....	289
8.3.2	RDREC und WRREC .....	290
8.3.3	RALRM .....	293
8.3.4	STATUS-Parameter für RDREC, WRREC und RALRM .....	296
8.3.5	DPRD_DAT und DPWR_DAT .....	300
8.3.6	DPNRM_DG .....	302
8.4	Alarme .....	305
8.4.1	Operationen ATTACH und DETACH .....	305
8.4.2	Weckalarme .....	308
8.4.2.1	SET_CINT (Weckalarm setzen) .....	308
8.4.2.2	QRY_CINT (Weckalarm abfragen) .....	310
8.4.3	Verzögerungsalarme .....	312
8.4.4	Asynchronereignisalarms .....	314
8.5	Diagnose (PROFINET oder PROFIBUS) .....	315
8.5.1	Diagnoseanweisungen .....	315
8.5.2	Diagnoseereignisse für die dezentrale Peripherie .....	315
8.5.3	Anweisung LED .....	316
8.5.4	Anweisung DeviceStates .....	318
8.5.5	Anweisung ModuleStates .....	319
8.5.6	Anweisung GET_DIAG .....	321
8.6	Impuls .....	328

8.6.1	Operation CTRL_PWM .....	328
8.6.2	Funktionsweise der Impulsausgänge.....	330
8.6.3	Konfigurieren eines Impulskanals für PWM.....	332
8.7	Datenprotokollierung .....	333
8.7.1	Datensatzstruktur der Datenprotokolle .....	334
8.7.2	Programmanweisungen zum Steuern von Datenprotokollen .....	335
8.7.2.1	DataLogCreate .....	335
8.7.2.2	DataLogOpen .....	338
8.7.2.3	DataLogClose .....	340
8.7.2.4	DataLogWrite .....	341
8.7.2.5	DataLogNewFile.....	342
8.7.3	Arbeiten mit Datenprotokollen.....	344
8.7.4	Grenzwerte für die Größe von Datenprotokolldateien .....	346
8.7.5	Beispielprogramm für Datenprotokolle.....	349
8.8	Datenbausteinststeuerung .....	353
8.8.1	READ_DBL, WRIT_DBL (Aus DB im Ladespeicher lesen, In DB im Ladespeicher schreiben).....	353
8.9	Gemeinsame Fehlercodes für die erweiterten Anweisungen .....	356
<b>9</b>	<b>Technologieanweisungen .....</b>	<b>357</b>
9.1	Schneller Zähler .....	357
9.1.1	Funktionsweise von schnellen Zählern .....	359
9.1.2	Konfiguration eines schnellen Zählers .....	365
9.2	PID-Regelung.....	366
9.2.1	Anweisung PID und Technologieobjekt einfügen .....	368
9.2.2	Operation PID_Compact .....	370
9.2.3	ErrorBit-Parameter der Anweisung PID_Compact.....	375
9.2.4	Anweisung PID_3STEP .....	376
9.2.5	ErrorBit-Parameter der Anweisung PID_3STEP .....	382
9.2.6	PID-Regler konfigurieren.....	384
9.2.7	Inbetriebnahme des PID-Reglers.....	386
9.3	Bewegungssteuerung .....	387
9.3.1	Achse konfigurieren .....	391
9.3.2	TO-Befehlstabelle PTO konfigurieren .....	394
9.3.3	Bewegungssteuerungsanweisungen .....	397
9.3.3.1	Anweisung MC_Power .....	397
9.3.3.2	Anweisung MC_Reset.....	400
9.3.3.3	Anweisung MC_Home.....	401
9.3.3.4	Anweisung MC_Halt.....	404
9.3.3.5	Anweisung MC_MoveAbsolute .....	406
9.3.3.6	Anweisung MC_MoveRelative .....	408
9.3.3.7	Anweisung MC_MoveVelocity.....	410
9.3.3.8	Anweisung MC_MoveJog.....	413
9.3.3.9	Anweisung MC_CommandTable .....	415
9.3.3.10	MC_ChangeDynamic .....	418
9.3.4	Funktionsweise der Bewegungssteuerung bei der S7-1200 .....	420
9.3.4.1	Für die Bewegungssteuerung verwendete CPU-Ausgänge .....	420
9.3.4.2	Hardware- und Software-Endschalter für die Bewegungssteuerung.....	421
9.3.4.3	Referenzpunktfahrt.....	425
9.3.4.4	Ruckbegrenzung .....	431

9.3.5	Inbetriebnahme .....	432
9.3.6	Aktive Befehle überwachen .....	435
9.3.6.1	Bewegungssteuerungsanweisungen mit einem Ausgangsparameter "Done" überwachen.....	435
9.3.6.2	Anweisung MC_Velocity überwachen.....	440
9.3.6.3	Anweisung MC_MoveJog überwachen.....	444
<b>10</b>	<b>Kommunikation.....</b>	<b>449</b>
10.1	Anzahl der unterstützten asynchronen Kommunikationsverbindungen .....	451
10.2	PROFINET .....	451
10.2.1	Verbindung zwischen lokaler und Partner-CPU .....	451
10.2.2	Offene Benutzerkommunikation.....	454
10.2.2.1	Verbindungs-IDs für die PROFINET-Anweisungen .....	454
10.2.2.2	Protokolle .....	457
10.2.2.3	Ad-hoc-Modus.....	458
10.2.2.4	TCP und ISO on TCP .....	459
10.2.2.5	UDP.....	474
10.2.2.6	T_CONFIG .....	479
10.2.2.7	Gemeinsame Parameter für Anweisungen.....	485
10.2.3	Kommunikation mit einem Programmiergerät .....	487
10.2.3.1	Hardware-Kommunikationsverbindung herstellen .....	487
10.2.3.2	Konfigurieren der Geräte .....	488
10.2.3.3	IP-Adressen zuweisen .....	489
10.2.3.4	Testen Ihres PROFINET-Netzwerks.....	489
10.2.4	Kommunikation HMI/PLC.....	489
10.2.4.1	Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten.....	491
10.2.5	Kommunikation PLC/PLC .....	491
10.2.5.1	Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten.....	492
10.2.5.2	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren.....	492
10.2.5.3	Sende- und Empfangsparameter konfigurieren.....	493
10.2.6	CPU und PROFINET IO-Gerät konfigurieren .....	495
10.2.7	Diagnose.....	498
10.2.8	Anweisungen für die dezentrale E/A.....	499
10.2.9	Diagnoseanweisungen.....	499
10.2.10	Diagnoseereignisse für die dezentrale Peripherie .....	499
10.3	PROFIBUS.....	500
10.3.1	Kommunikationsmodule für PROFIBUS.....	501
10.3.1.1	Anbindung an PROFIBUS .....	501
10.3.1.2	Kommunikationsdienste der PROFIBUS-CMs .....	502
10.3.1.3	Weitere Eigenschaften der PROFIBUS-CMs .....	503
10.3.1.4	Konfigurationsbeispiele für PROFIBUS .....	505
10.3.2	DP-Master und -Slave konfigurieren.....	506
10.3.2.1	CM 1243-5 (DP-Master) und DP-Slave hinzufügen.....	506
10.3.2.2	Konfigurieren der logischen Netzwerkverbindungen zwischen zwei PROFIBUS-Geräten .....	506
10.3.2.3	PROFIBUS-Adressen zum CM 1243-5 und DP-Slave zuweisen .....	507
10.3.3	Anweisungen für die dezentrale E/A.....	508
10.3.4	Diagnoseanweisungen.....	509
10.3.5	Diagnoseereignisse für die dezentrale Peripherie .....	509
10.4	ASi.....	509
10.4.1	AS-i-Master und -Slavegeräte konfigurieren.....	510
10.4.1.1	AS-i-Mastermodul CM 1243-2 und AS-i-Slave hinzufügen .....	510
10.4.1.2	Logische Netzwerkverbindungen zwischen zwei AS-i-Geräten konfigurieren .....	511

10.4.1.3	Eigenschaften des AS-i-Masters CM1243-2 konfigurieren.....	511
10.4.1.4	Einem AS-i-Slave eine AS-i-Adresse zuweisen.....	512
10.4.2	Datenaustausch zwischen dem Anwenderprogramm und AS-i-Slaves .....	513
10.4.2.1	STEP 7 Basic konfigurieren .....	513
10.4.2.2	Slaves mit STEP 7 konfigurieren .....	514
10.4.3	Anweisungen für die dezentrale E/A.....	516
10.4.4	Mit AS-i-Online-Werkzeugen arbeiten .....	516
10.5	S7-Kommunikation.....	518
10.5.1	Anweisungen GET und PUT .....	518
10.5.2	S7-Verbindung erstellen.....	522
10.5.3	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren.....	522
10.5.4	Zuweisung von Verbindungsparametern für GET/PUT .....	523
10.5.4.1	Verbindungsparameter.....	523
10.5.4.2	S7-Verbindung von CPU zu CPU konfigurieren .....	526
<b>11</b>	<b>Webserver .....</b>	<b>531</b>
11.1	Webserver aktivieren .....	532
11.2	Standard-Webseiten .....	533
11.2.1	Über den PC auf die Standard-Webseiten zugreifen.....	533
11.2.2	Aufbau der Standard-Webseiten.....	535
11.2.3	Einleitung.....	537
11.2.4	Start.....	538
11.2.5	Identifikation .....	539
11.2.6	Diagnosepuffer.....	540
11.2.7	Modulinformationen.....	540
11.2.8	Kommunikation .....	543
11.2.9	Variablenzustand .....	544
11.2.10	Datenprotokolle .....	546
11.2.11	Firmware aktualisieren.....	549
11.3	Benutzerdefinierte Webseiten .....	551
11.3.1	HTML-Seiten anlegen .....	552
11.3.2	Vom S7-1200 Webserver unterstützte AWP-Befehle .....	553
11.3.2.1	Variablen lesen .....	554
11.3.2.2	Variablen schreiben .....	555
11.3.2.3	Sondervariablen lesen .....	557
11.3.2.4	Sondervariablen schreiben .....	559
11.3.2.5	Alias für einen Variablenverweis nutzen .....	560
11.3.2.6	Enum-Typen definieren.....	561
11.3.2.7	CPU-Variablen mit einem Enum-Typ referenzieren .....	562
11.3.2.8	Fragmente erstellen .....	563
11.3.2.9	Fragmente importieren.....	564
11.3.2.10	Definitionen verbinden .....	565
11.3.2.11	Handhabung von Variablennamen mit Sonderzeichen .....	565
11.3.3	Verwendung von benutzerdefinierten Webseiten konfigurieren .....	567
11.3.4	WWW-Anweisung für benutzerdefinierte Webseiten programmieren .....	569
11.3.5	Programmbausteine in die CPU laden.....	571
11.3.6	Über den PC auf die benutzerdefinierten Webseiten zugreifen.....	571
11.3.7	Einschränkungen bei benutzerdefinierten Webseiten .....	571
11.3.8	Beispiel für eine benutzerdefinierte Webseite.....	572
11.3.8.1	Webseite zum Beobachten und Steuern einer Windturbine .....	572
11.3.8.2	Steuerungsdaten lesen und anzeigen .....	575

11.3.8.3	Enum-Typ verwenden.....	575
11.3.8.4	Benutzereingaben in die Steuerung schreiben.....	576
11.3.8.5	Sondervariablen schreiben.....	578
11.3.8.6	Referenz: HTML-Code der Webseite "Remote Wind Turbine Monitor".....	578
11.3.8.7	Konfiguration der Beispiel-Webseite in STEP 7.....	582
11.3.9	Benutzerdefinierte Webseiten in mehreren Sprachen einrichten.....	583
11.3.9.1	Ordnerstruktur anlegen.....	584
11.3.9.2	Sprachumschaltung programmieren.....	584
11.3.9.3	STEP 7 für die Verwendung einer mehrsprachigen Seitenstruktur konfigurieren.....	587
11.3.10	Erweiterte Steuerung von benutzerdefinierten Webseiten.....	587
11.4	Einschränkungen.....	591
11.4.1	Eingeschränkte Funktionen bei deaktiviertem JavaScript.....	592
11.4.2	Eingeschränkte Funktionen, wenn keine Cookies zugelassen sind.....	593
11.4.3	Siemens-Sicherheitszertifikat importieren.....	593
11.4.4	Datenprotokolle im CSV-Format in nicht amerikanische/englische Versionen von Microsoft Excel importieren.....	595
<b>12</b>	<b>Kommunikationsprozessor.....</b>	<b>597</b>
12.1	Mit den seriellen Kommunikationsschnittstellen arbeiten.....	597
12.2	Abschließen eines RS485-Busanschlusssteckers.....	598
12.3	Punkt-zu-Punkt-Kommunikation (PtP).....	599
12.3.1	Punkt-zu-Punkt-Operationen.....	600
12.3.1.1	Gemeinsame Parameter für Punkt-zu-Punkt-Operationen.....	600
12.3.1.2	Operation PORT_CFG.....	602
12.3.1.3	Operation SEND_CFG.....	604
12.3.1.4	Operation RCV_CFG.....	605
12.3.1.5	Operation SEND_PTP.....	610
12.3.1.6	Operation RCV_PTP.....	614
12.3.1.7	Operation RCV_RST.....	615
12.3.1.8	Operation SGN_GET.....	616
12.3.1.9	Operation SGN_SET.....	617
12.3.2	Kommunikationsanschlüsse konfigurieren.....	618
12.3.2.1	Steuerung der Flusskontrolle.....	620
12.3.3	Sende- und Empfangsparameter konfigurieren.....	622
12.3.3.1	Sendeparameter konfigurieren.....	622
12.3.3.2	Empfangsparameter konfigurieren.....	623
12.3.4	Programmieren der PtP-Kommunikation.....	631
12.3.4.1	Abfragearchitektur.....	632
12.3.5	Beispiel: Punkt-zu-Punkt-Kommunikation.....	633
12.3.5.1	Kommunikationsmodul konfigurieren.....	634
12.3.5.2	RS422 und RS485 konfigurieren.....	637
12.3.5.3	STEP 7-Programm programmieren.....	640
12.3.5.4	Terminalemulator konfigurieren.....	641
12.3.5.5	Beispielprogramm ausführen.....	642
12.4	Kommunikation über die universelle serielle Schnittstelle (USS).....	642
12.4.1	Voraussetzungen für den Einsatz des USS-Protokolls.....	644
12.4.2	Operation USS_DRV.....	647
12.4.3	Operation USS_PORT.....	650
12.4.4	Operation USS_RPM.....	651
12.4.5	Operation USS_WPM.....	652

12.4.6	USS-Zustandscodes .....	654
12.4.7	Allgemeine Informationen zur Antriebseinrichtung .....	656
12.5	Modbus-Kommunikation .....	659
12.5.1	Überblick über die Modbus RTU- und TCP-Kommunikation .....	659
12.5.2	Modbus TCP .....	662
12.5.2.1	MB_CLIENT (Modbus TCP).....	662
12.5.2.2	MB_SERVER (Modbus TCP).....	669
12.5.2.3	Beispiel MB_SERVER: Mehrere TCP-Verbindungen .....	674
12.5.2.4	Beispiel 1 für MB_CLIENT: Mehrere Anforderungen mit gemeinsamer TCP-Verbindung .....	675
12.5.2.5	Beispiel 2 für MB_CLIENT: Mehrere Anforderungen mit unterschiedlichen TCP-Verbindungen.....	676
12.5.2.6	Beispiel 3 für MB_CLIENT: Schreibanforderung für das Prozessabbild der Ausgänge .....	677
12.5.2.7	Beispiel 4 für MB_CLIENT: Mehrere Anforderungen koordinieren.....	677
12.5.3	Modbus RTU .....	678
12.5.3.1	MB_COMM_LOAD .....	679
12.5.3.2	MB_MASTER .....	682
12.5.3.3	MB_SLAVE .....	689
12.5.3.4	Beispielprogramm für einen Modbus RTU-Master .....	695
12.5.3.5	Beispielprogramm für einen Modbus RTU-Slave .....	697
12.6	Telecontrol und TeleService mit dem CP 1242-7 .....	698
12.6.1	Anschluss an ein GSM-Netz .....	698
12.6.2	Anwendungen des CP 1242-7 .....	699
12.6.3	Weitere Eigenschaften des CP .....	701
12.6.4	Zubehör.....	702
12.6.5	Konfigurationsbeispiele für Telecontrol.....	703
<b>13</b>	<b>Teleservice-Kommunikation (SMTP-E-Mail) .....</b>	<b>709</b>
13.1	E-Mail-Anweisung TM_Mail-Übertragung .....	709
<b>14</b>	<b>Online- und Diagnose-Tools .....</b>	<b>717</b>
14.1	Status-LEDs .....	717
14.2	Online-Verbindung mit einer CPU herstellen .....	719
14.3	Einem PROFINET IO-Gerät online einen Namen zuweisen .....	720
14.4	Einstellen der IP-Adresse und der Uhrzeit.....	721
14.5	Auf Werkseinstellungen zurücksetzen .....	722
14.6	Bedienpanel für die Online-CPU.....	723
14.7	Überwachung von Zykluszeit und Speicherauslastung .....	723
14.8	Diagnoseereignisse in der CPU anzeigen .....	724
14.9	Vergleichen von Offline- und Online-CPUs.....	725
14.10	Werte in der CPU beobachten und steuern .....	726
14.10.1	Online gehen, um die Werte in der CPU zu beobachten .....	727
14.10.2	Zustand im Programmiereditor anzeigen.....	728
14.10.3	Online-Werte eines DBs erfassen, um die Startwerte zurückzusetzen .....	728
14.10.4	Werte in der CPU über die Beobachtungstabelle beobachten und steuern .....	729
14.10.4.1	Variablen mit Trigger beobachten oder steuern .....	731
14.10.4.2	Ausgänge im Betriebszustand STOP freischalten.....	732
14.10.5	Werte in der CPU forcen .....	732

14.10.5.1	Arbeiten mit der Forcetablelle.....	732
14.10.5.2	Funktionsweise der Forcefunktion .....	733
14.11	Laden im Betriebszustand RUN.....	735
14.11.1	Voraussetzungen für "Laden im Betriebszustand RUN".....	736
14.11.2	Ändern des Programms im Betriebszustand RUN .....	736
14.11.3	Ausgewählte Bausteine laden.....	738
14.11.4	Einen einzelnen ausgewählten Baustein mit einem Übersetzungsfehler in einem anderen Baustein laden .....	739
14.11.5	Systemreaktion bei fehlgeschlagenem Ladevorgang .....	740
14.11.6	Laden des Programms im Betriebszustand RUN .....	741
<b>A</b>	<b>Technische Daten.....</b>	<b>743</b>
A.1	Allgemeine technische Daten .....	743
A.2	CPU 1211C.....	750
A.2.1	Allgemeine technische Daten und Leistungsmerkmale.....	750
A.2.2	Von der CPU 1211C unterstützte Zeiten, Zähler und Codebausteine.....	751
A.2.3	Digitale Eingänge und Ausgänge .....	753
A.2.4	Analoge Eingänge.....	755
A.2.4.1	Schrittantwort der integrierten analogen Eingänge der CPU.....	756
A.2.4.2	Abtastzeit der integrierten analogen Ports der CPU.....	756
A.2.5	Schaltpläne der CPU 1211C.....	757
A.3	CPU 1212C.....	760
A.3.1	Allgemeine technische Daten und Leistungsmerkmale.....	760
A.3.2	Von der CPU 1212C unterstützte Zeiten, Zähler und Codebausteine.....	761
A.3.3	Digitale Eingänge und Ausgänge .....	764
A.3.4	Analoge Eingänge.....	765
A.3.4.1	Schrittantwort der integrierten analogen Eingänge der CPU.....	766
A.3.4.2	Abtastzeit der integrierten analogen Ports der CPU.....	766
A.3.5	Schaltpläne der CPU 1212C.....	767
A.4	CPU 1214C.....	770
A.4.1	Allgemeine technische Daten und Leistungsmerkmale.....	770
A.4.2	Von der CPU 1214C unterstützte Zeiten, Zähler und Codebausteine.....	771
A.4.3	Digitale Eingänge und Ausgänge .....	774
A.4.4	Analoge Eingänge.....	775
A.4.4.1	Schrittantwort der integrierten analogen Eingänge der CPU.....	776
A.4.4.2	Abtastzeit der integrierten analogen Ports der CPU.....	776
A.4.5	Schaltpläne der CPU 1214C.....	777
A.5	CPU 1215C.....	780
A.5.1	Allgemeine technische Daten und Leistungsmerkmale.....	780
A.5.2	Von der CPU 1215C unterstützte Zeiten, Zähler und Codebausteine.....	782
A.5.3	Digitale Eingänge und Ausgänge .....	784
A.5.4	Analogeingänge und -ausgänge.....	786
A.5.4.1	Technische Daten der analogen Eingänge.....	786
A.5.4.2	Schrittantwort der integrierten analogen Eingänge der CPU.....	786
A.5.4.3	Abtastzeit der integrierten analogen Ports der CPU.....	787
A.5.4.4	Technische Daten der Analogausgänge.....	787
A.5.5	Schaltpläne der CPU 1215C.....	788
A.6	Digitale Signalmodule (SM) .....	792
A.6.1	Technische Daten für das digitale Eingangsmodul SM 1221 .....	792

A.6.2	Technische Daten für das digitale Ausgangsmodul SM 1222 mit 8 Ausgängen .....	794
A.6.3	Technische Daten für das digitale Ausgangsmodul SM 1222 mit 16 Ausgängen .....	796
A.6.4	Technische Daten des digitalen Ein-/Ausgangsmoduls SM 1223 (V DC) .....	800
A.6.5	Technische Daten des digitalen Ein-/Ausgangsmoduls SM 1223 (AC).....	805
A.7	Analoge Signalmodule (SM) .....	808
A.7.1	Technische Daten des SM 1231 Analogeingabemoduls .....	808
A.7.2	Technische Daten des SM 1232 Analogausgabemoduls .....	812
A.7.3	Technische Daten des SM 1234 Analogein-/Analogausgabemoduls.....	814
A.7.4	Schrittantwort der analogen Eingänge.....	817
A.7.5	Abtastzeit und Aktualisierungszeiten der Analogeingänge.....	817
A.7.6	Messbereiche der analogen Eingänge für Spannung.....	818
A.7.7	Messbereiche der analogen Eingänge für Strom .....	818
A.7.8	Ausgangsmessbereiche (AO) für Spannung und Strom (SB und SM).....	819
A.8	Thermoelement- und RTD-Signalmodule (SMs).....	820
A.8.1	SM 1231 Thermoelement.....	820
A.8.1.1	Grundlegende Funktionsweise eines Thermoelements.....	823
A.8.1.2	Auswahltabellen für das SM 1231 Thermoelement .....	824
A.8.2	SM 1231 RTD .....	826
A.8.2.1	Auswahltabellen für das SM 1231 RTD .....	829
A.9	Digitale Signalboards (SBs) .....	832
A.9.1	Technische Daten des SB 1221 200 kHz Digitaleingabe .....	832
A.9.2	Technische Daten des SB 1222 200 kHz Digitalausgabe .....	834
A.9.3	Technische Daten des SB 1223 200 kHz Digitalein-/Digitalausgabe .....	837
A.9.4	Technische Daten SB 1223 2 x 24 V DC Eingang / 2 x 24 V DC Ausgang.....	840
A.10	Analoge Signalboards (SBs).....	842
A.10.1	Technische Daten des SB 1231 1 Analogeingang .....	842
A.10.2	Technische Daten des SB 1232 1 Analogausgabe .....	845
A.10.3	Messbereiche der analogen Eingänge und Ausgänge .....	847
A.10.3.1	Schrittantwort der analogen Eingänge.....	847
A.10.3.2	Abtastzeit und Aktualisierungszeiten der Analogeingänge.....	847
A.10.3.3	Messbereiche der analogen Eingänge für Spannung.....	848
A.10.3.4	Messbereiche der analogen Eingänge für Strom .....	848
A.10.3.5	Ausgangsmessbereiche (AO) für Spannung und Strom (SB und SM).....	849
A.10.4	Thermoelement-SBs .....	850
A.10.4.1	Technische Daten des SB 1231 1 Analogeingang Thermoelement.....	850
A.10.4.2	Grundlegende Funktionsweise eines Thermoelements.....	851
A.10.5	RTD-SBs .....	854
A.10.5.1	Technische Daten des SB 1231 1 Analogeingang RTD.....	854
A.10.5.2	Auswahltabellen für das SB 1231 RTD.....	857
A.11	BB 1297 Batterieboard.....	859
A.12	Kommunikationsschnittstellen.....	861
A.12.1	PROFIBUS .....	861
A.12.1.1	CM 1242-5.....	861
A.12.1.2	CM 1243-5.....	862
A.12.2	GPRS .....	864
A.12.2.1	CP 1242-7 .....	864
A.12.3	CM 1243-2 AS-i-Master .....	867
A.12.3.1	Technische Daten des AS-i Master CM 1243-2.....	867
A.12.3.2	Elektrische Anschlüsse des AS-i Masters CM 1243-2.....	868



A.12.4	RS232, RS422 und RS485 .....	869
A.12.4.1	Technische Daten des CM 1241 RS485 .....	869
A.12.4.2	Technische Daten für CM 1241 RS232 .....	872
A.12.4.3	Technische Daten des CM 1241 RS422/485 .....	873
A.13	TeleService (TS-Adapter und TS-Adaptermodul).....	874
A.14	SIMATIC Memory Cards .....	875
A.15	Eingangssimulatoren .....	875
A.16	Steckleitung für Erweiterungsmodule .....	877
A.17	Zugehörige Produkte .....	877
A.17.1	PM 1207 Stromversorgungsmodul .....	877
A.17.2	CSM 1277 Compact Switch Module .....	878
<b>B</b>	<b>Berechnen der Leistungsbilanz.....</b>	<b>879</b>
<b>C</b>	<b>Bestellnummern.....</b>	<b>883</b>
C.1	CPU-Module.....	883
C.2	Signalmodule (SMs), Signalboards (SBs) und Batterieboards (BBs).....	883
C.3	Kommunikation .....	885
C.4	Sonstige Module .....	886
C.5	Memory Cards.....	886
C.6	Grundlegende HMI-Geräte .....	886
C.7	Ersatzteile und sonstige Hardware .....	887
C.8	Programmiersoftware.....	887
C.9	Dokumentation .....	888
	<b>Index.....</b>	<b>889</b>



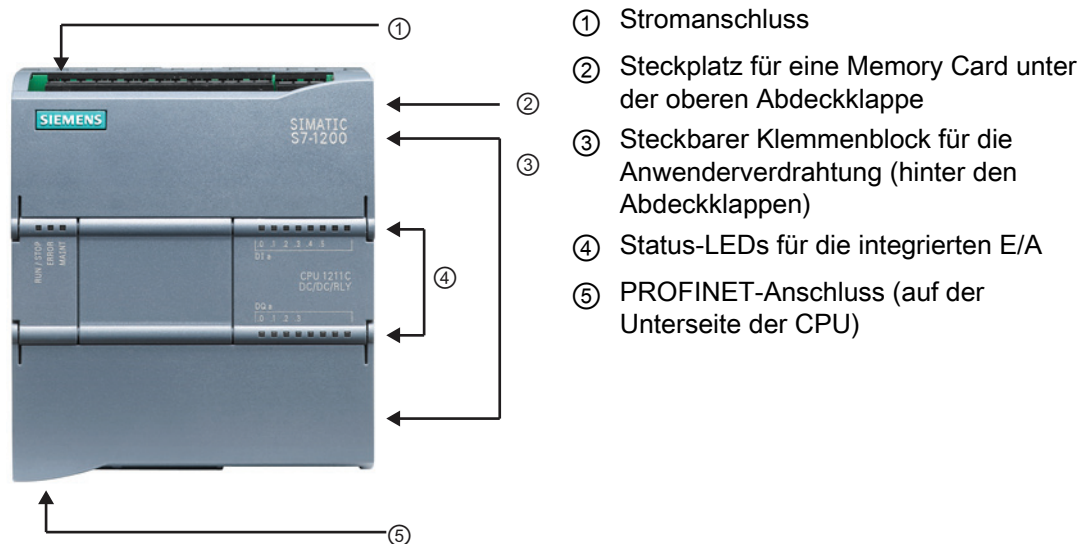
## Produktübersicht

### 1.1 Einführung in die S7-1200 SPS

Die Steuerung S7-1200 bietet Ihnen die erforderliche Flexibilität und Leistung zur Steuerung einer breiten Palette von Geräten für Ihre Automatisierungslösungen. Durch das kompakte Design, die flexible Konfiguration und einen leistungsstarken Befehlssatz eignet sich die S7-1200 hervorragend für eine große Bandbreite von Steuerungsanwendungen.

Die CPU umfasst einen Mikroprozessor, eine integrierte Spannungsversorgung, Eingangskreise und Ausgangskreise, integriertes PROFINET, Peripherie zur Bewegungssteuerung in Hochgeschwindigkeit sowie integrierte Analogeingänge in einem kompakten Gehäuse und bildet somit eine leistungsstarke Steuerung. Nachdem Sie Ihr Programm geladen haben, enthält die CPU die erforderliche Logik, damit Sie die Geräte in Ihrer Anwendung beobachten und steuern können. Die CPU beobachtet Eingänge und ändert Ausgänge anhand der Befehle Ihres Anwenderprogramms, das Boolesche Verknüpfungen, Zähl- und Zeitfunktionen, komplexe arithmetische Operationen und Kommunikation mit anderen intelligenten Geräten umfassen kann.

Die CPU verfügt über einen PROFINET-Port zur Kommunikation über ein PROFINET-Netzwerk. Für die Kommunikation über PROFIBUS-, GPRS-, RS485- oder RS232-Netzwerke stehen weitere Module zur Verfügung.



1.1 Einführung in die S7-1200 SPS

Verschiedene Sicherheitsfunktionen schützen den Zugriff auf die CPU und das Steuerungsprogramm:

- Jede CPU ist mit einem Passwortschutz (Seite 176) ausgestattet, mit dem der Zugriff auf die CPU-Funktionen nach Bedarf eingerichtet werden kann.
- Sie können mit dem Knowhow-Schutz (Seite 177) den Code in einem bestimmten Baustein verbergen.
- Mit dem Kopierschutz (Seite 178) können Sie Ihr Programm mit einer bestimmten Memory Card oder CPU verknüpfen.

Tabelle 1- 1 Vergleich der CPU-Varianten

Merkmal		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
Abmessungen (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75
Anwenderspeiche r	Arbeitsspeicher	30 KB	50 KB	75 KB	100 KB
	Ladespeicher	1 MB	1 MB	4 MB	4 MB
	Remanent	10 KB	10 KB	10 KB	10 KB
Integrierte E/A	Digital	6 Eingänge/4 Ausgänge	8 Eingänge/6 Ausgänge	14 Eingänge/10 Ausgänge	14 Eingänge/10 Ausgänge
	Analog	2 Eingänge	2 Eingänge	2 Eingänge	2 Eingänge/2 Ausgänge
Größe des Prozessabbilds	Eingänge (E)	1024 Byte	1024 Byte	1024 Byte	1024 Byte
	Ausgang (A)	1024 Byte	1024 Byte	1024 Byte	1024 Byte
Merker (M)		4096 Byte	4096 Byte	8192 Byte	8192 Byte
Erweiterung: Signalmodul (SM)		Keine	2	8	8
Signalboard (SB), Batterieboard (BB) oder Kommunikationsboard (CB)		1	1	1	1
Kommunikationsmodule (CM) (Anbau links)		3	3	3	3
Schnelle Zähler	Summe	3 integrierte E/A, 5 mit SB	4 integrierte E/A, 6 mit SB	6	6
	Einphasenzähler	3 bei 100 kHz SB: 2 bei 30 kHz	3 bei 100 kHz 1 bei 30 kHz SB: 2 bei 30 kHz	3 bei 100 kHz 3 bei 30 kHz	3 bei 100 kHz 3 bei 30 kHz
	A/B-Zähler	3 bei 80 kHz SB: 2 bei 20 kHz	3 bei 80 kHz 1 bei 20 kHz SB: 2 bei 20 kHz	3 bei 80 kHz 3 bei 20 kHz	3 bei 80 kHz 3 bei 20 kHz
Impulsausgänge <sup>1</sup>		4	4	4	4
Memory Card		SIMATIC Memory Card (optional)			
Pufferung Echtzeituhr		typ. 20 Tage / min. 12 Tage bei 40 Grad C (wartungsfreier Hochleistungskondensator)			
PROFINET		1 Ethernet-Kommunikationsport			2 Ethernet-Kommunikationsports

Merkmal	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
Ausführungszeit arithm. Operationen	2,3 µs/Operation			
Ausführungszeit Boolesche Operationen	0,08 µs/Operation			

<sup>1</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Jede CPU bietet zweckbestimmte HMI-Verbindungen, um bis zu 3 HMI-Geräte zu unterstützen. Wie viele HMI-Geräte insgesamt unterstützt werden, hängt von den Typen der HMI-Panels in Ihrer Konfiguration ab. Sie können beispielsweise bis zu drei SIMATIC Basic Panels an Ihre CPU anschließen, oder Sie können bis zu zwei SIMATIC Comfort Panels und ein zusätzliches Basic Panel anschließen.

Die verschiedenen CPU-Ausführungen bieten eine Vielfalt an Leistungsmerkmalen und Funktionen, damit Sie effektive Lösungen für verschiedenste Anwendungen erstellen können. Ausführliche Informationen zu bestimmten CPUs finden Sie in den technischen Daten (Seite 743).

Tabelle 1-2 Von der S7-1200 unterstützte Bausteine, Zeiten und Zähler

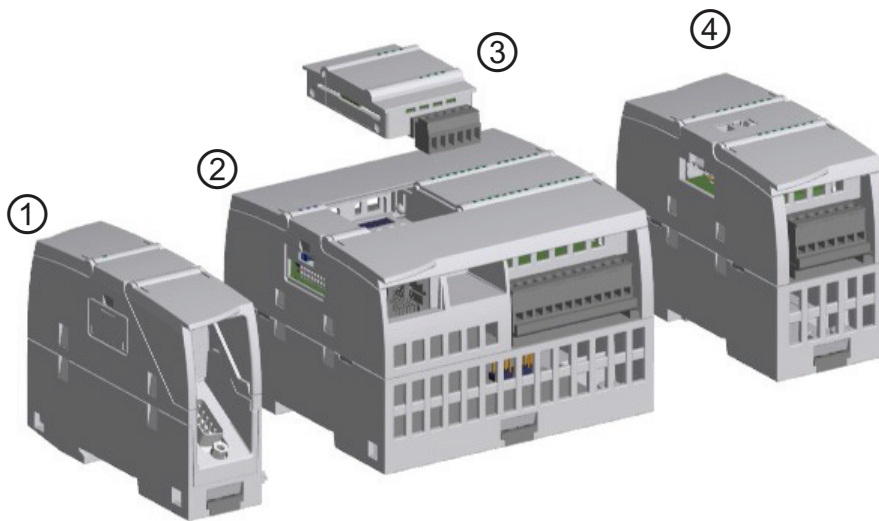
Element	Beschreibung	
Bausteine	Typ	OB, FB, FC, DB
	Größe	30 KB (CPU 1211C) 50 KB (CPU 1212C) 64 KB (CPU 1214C und CPU 1215C)
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	1 bis 65535 (z. B. FB 1 bis FB 65535)
	Schachteltiefe	16 aus Zyklus- oder Anlauf-OBs, 4 aus Verzögerungsalarm-, Uhrzeitalarm-, Weckalarm-, Prozessalarm-, Zeitfehler- oder Diagnosefehler-OBs
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.
	OBs	Programmzyklus
Anlauf		Mehrere: OB 100, OB 200 bis OB 65535
Verzögerungsalarme und Weckalarme		4 <sup>1</sup> (1 pro Ereignis): OB 200 bis OB 65535
Prozessalarme (Flanken und HSC)		50 (1 pro Ereignis): OB 200 bis OB 65535
Zeitfehleralarme		1: OB 80
Diagnosefehleralarme		1: OB 82
Zeiten	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Bytes pro Zeit
Zähler	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt

Element	Beschreibung
Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"><li>• SInt, USInt: 3 Byte</li><li>• Int, UInt: 6 Byte</li><li>• DInt, UDInt: 12 Byte</li></ul>

<sup>1</sup> Verzögerungs- und Weckalarme nutzen dieselben Ressourcen in der CPU. Es darf insgesamt maximal 4 dieser Alarme geben (Verzögerungs- plus Weckalarme). 4 Verzögerungsalarme und 4 Weckalarme sind nicht möglich.

## 1.2 Erweiterung der CPU-Funktionen

Die Produktfamilie S7-1200 bietet eine Vielzahl von Modulen und steckbaren Boards zur Erweiterung der CPU um zusätzliche E/A oder andere Kommunikationsprotokolle. Ausführliche Informationen zu bestimmten Modulen finden Sie in den technischen Daten (Seite 743).



- ① Kommunikationsmodul (CM), Kommunikationsprozessor (CP) oder TS-Adapter
- ② CPU
- ③ Signalboard (SB), Kommunikationsboard (CB) oder Batterieboard (BB)
- ④ Signalmodul (SM)

Tabelle 1- 3 Digitale Signalmodule und Signalboards

Typ	Nur Eingang	Nur Ausgang	Ein-/Ausgang kombiniert
③ digitales SB	<ul style="list-style-type: none"> <li>• 4 x 24-V-DC-Eingänge, 200 kHz</li> <li>• 4 x 5-V-DC-Eingänge, 200 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 4 x 24-V-DC-Ausgänge, 200 kHz</li> <li>• 4 x 5-V-DC-Ausgänge, 200 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 2 x 24-V-DC-Eingänge / 2 x 24-V-DC-Ausgänge</li> <li>• 2 x 24-V-DC-Eingänge / 2 x 24-V-DC-Ausgänge, 200 kHz</li> <li>• 2 x 5-V-DC-Eingänge / 2 x 5-V-DC-Ausgänge, 200 kHz</li> </ul>
④ digitales SM	<ul style="list-style-type: none"> <li>• 8 x 24-V-DC-Eingänge</li> </ul>	<ul style="list-style-type: none"> <li>• 8 x 24-V-DC-Ausgänge</li> <li>• 8 x Relaisausgänge</li> <li>• 8 x Relaisausgang (Umschaltung)</li> </ul>	<ul style="list-style-type: none"> <li>• 8 x 24-V-DC-Eingänge / 8 x 24-V-DC-Ausgänge</li> <li>• 8 x 24-V-DC-Eingänge / 8 x Relaisausgänge</li> <li>• 8 x 120/230-V-AC-Eingänge / 8 x Relaisausgänge</li> </ul>
	<ul style="list-style-type: none"> <li>• 16 x 24-V-DC-Eingänge</li> </ul>	<ul style="list-style-type: none"> <li>• 16 x 24-V-DC-Ausgänge</li> <li>• 16 x Relaisausgänge</li> </ul>	<ul style="list-style-type: none"> <li>• 16 x 24-V-DC-Eingänge / 16 x 24-V-DC-Ausgänge</li> <li>• 16 x 24-V-DC-Eingänge / 16 x Relaisausgänge</li> </ul>

Tabelle 1- 4 Analoge Signalmodule und Signalboards

Typ	Nur Eingang	Nur Ausgang	Ein-/Ausgang kombiniert
③ analoges SB	<ul style="list-style-type: none"> <li>• 1 x 12-Bit-Analogeingang</li> <li>• 1 x 16-Bit-RTD</li> <li>• 1 x 16-Bit-Thermoelement</li> </ul>	<ul style="list-style-type: none"> <li>• 1 x Analogausgang</li> </ul>	-
④ analoges SM	<ul style="list-style-type: none"> <li>• 4 x Analogeingänge</li> <li>• 4 x Analogeingänge x 16 Bit</li> <li>• 8 x Analogeingänge</li> <li>• Thermoelement:                             <ul style="list-style-type: none"> <li>– 4 x 16-Bit-TC</li> <li>– 8 x 16-Bit-TC</li> </ul> </li> <li>• RTD:                             <ul style="list-style-type: none"> <li>– 4 x 16-Bit-RTD</li> <li>– 8 x 16-Bit-RTD</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 2 x Analogausgänge</li> <li>• 4 x Analogausgänge</li> </ul>	<ul style="list-style-type: none"> <li>• 4 x Analogeingänge/2 x Analogausgänge</li> </ul>

Tabelle 1- 5 Kommunikationsschnittstellen

Modul	Typ	Beschreibung
① Kommunikationsmodul (CM)	RS232	Vollduplex
	RS422/485	Vollduplex (RS422) Halbduplex (RS485)
	PROFIBUS-Master	DPV1
	PROFIBUS-Slave	DPV1
	AS-i-Master (CM 1243-2)	AS-Interface
① Kommunikationsprozessor (CP)	Modem anschließbar	GPRS
① Kommunikationsboard (CB)	RS485	Halbduplex
① TeleService	TS-Adapter IE Basic <sup>1</sup>	Anschluss an CPU
	TS-Adapter GSM	GSM/GPRS
	TS-Adapter Modem	Modem
	TS-Adapter ISDN	ISDN
	TS-Adapter RS232	RS232

<sup>1</sup> Mit dem TS-Adapter können Sie verschiedene Kommunikationsschnittstellen an den PROFINET-Port der CPU anschließen. Sie installieren den TS-Adapter an der linken Seite der CPU und stecken dann (bis zu 3) TS-Adaptermodule auf den TS-Adapter.

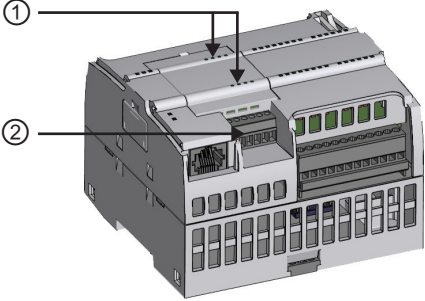
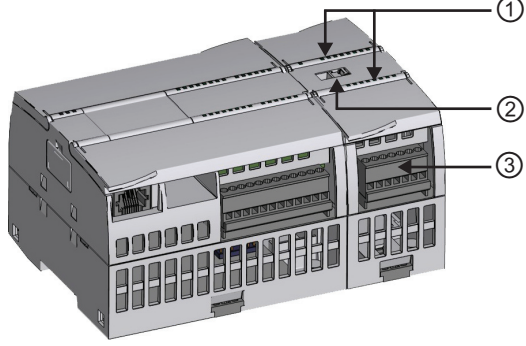
Tabelle 1- 6 Andere Boards

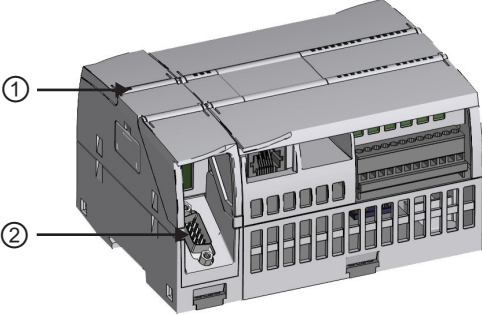
Modul	Beschreibung
③ Batterieboard	Wird in die Schnittstelle für Erweiterungsboards auf der Vorderseite der CPU gesteckt. Bietet langfristige Pufferung der Echtzeituhr.



## 1.3 S7-1200 Module

Tabelle 1-7 S7-1200 Erweiterungsmodule

Art des Moduls	Beschreibung	
<p>Die CPU unterstützt ein steckbares Erweiterungsboard:</p> <ul style="list-style-type: none"> <li>• Ein Signalboard (SB) stellt zusätzliche E/A für Ihre CPU bereit. Das SB wird auf der Vorderseite der CPU angeschlossen.</li> <li>• Mit einem Kommunikationsboard (CB) können Sie Ihre CPU um einen Kommunikationsanschluss erweitern.</li> <li>• Ein Batterieboard (BB) gewährleistet die langfristige Pufferung der Echtzeituhr.</li> </ul>		<p>① Status-LEDs am SB</p> <p>② Steckbarer Klemmenblock für die Anwenderverdrahtung</p>
<p>Signalmodule (SMs) erweitern die CPU um zusätzliche Funktionalität. SMs werden an der rechten Seite der CPU angeschlossen.</p> <ul style="list-style-type: none"> <li>• Digitale E/A</li> <li>• Analoge E/A</li> <li>• RTD und Thermoelement</li> </ul>		

Art des Moduls	Beschreibung					
<p>Kommunikationsmodule (CMs) und Kommunikationsprozessoren (CPs) erweitern die CPU um verschiedene Kommunikationsmöglichkeiten, z. B. PROFIBUS oder RS232/RS485 (für PtP, Modbus oder USS) oder AS-i-Master. Ein CP bietet Möglichkeiten für andere Arten der Kommunikation, z. B. für den Anschluss der CPU über ein GPRS-Netzwerk.</p> <ul style="list-style-type: none"> <li>Die CPU unterstützt maximal 3 CMs oder CPs.</li> <li>Jedes CM bzw. jeder CP wird an der linken Seite der CPU angeschlossen (bzw. an der linken Seite eines anderen CMs oder CPs).</li> </ul>		<table border="1"> <tr> <td data-bbox="1160 344 1230 389">①</td> <td data-bbox="1235 344 1439 389">Status-LEDs</td> </tr> <tr> <td data-bbox="1160 396 1230 441">②</td> <td data-bbox="1235 396 1439 441">Kommunikationsstecker</td> </tr> </table>	①	Status-LEDs	②	Kommunikationsstecker
①	Status-LEDs					
②	Kommunikationsstecker					

## 1.4 Neue Funktionen

Folgende Funktionen sind in dieser Version neu hinzugekommen:

- Eine Standardseite für den Webserver zur Durchführung eines CPU-Firmware-Updates (Seite 549)
- Die Möglichkeit, drei PROFIBUS DP CM 1243-5 Mastermodule oder drei AS-i CM 1243-2 Mastermodule zu verwenden

---

### Hinweis

Um drei AS-i-Module als Master zu verwenden, müssen Sie die Firmware der AS-i-Module aktualisieren.

---

### Neue Module für die S7-1200


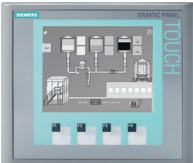
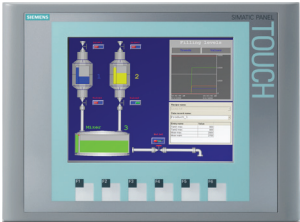
Eine Vielzahl neuer Module erweitert die Leistung der S7-1200 CPU und bietet Ihnen die erforderliche Flexibilität, um Ihre Automatisierungsanforderungen zu erfüllen:

- Neue, verbesserte CPUs:
  - Die neue CPU 1215C DC/DC/DC, CPU 1215C DC/DC/Relais und CPU 1215C AC/DC/Relais bieten 100 KB Arbeitsspeicher, Dual-Ethernet und Analogausgänge.
  - Die neue, verbesserte CPU 1211C, CPU 1212C und CPU 1214C verfügen über eine schnellere Verarbeitungszeit, die Möglichkeit für 4 PTOs (die CPU 1211C benötigt hierfür ein Signalboard), einen größeren remanenten Speicher (10 KB) und eine längere Haltezeit der Echtzeituhr (20 Tage).
- Neues E/A-Signalmodul: Das SM 1231 AI 4 x 16 Bit bietet eine höhere Abtastrate und eine größere Anzahl von Bits.
- Das neue Batterieboard (BB 1297) bietet langfristige Pufferung der Echtzeituhr. Das BB 1297 kann in den Steckplatz des Signalboards der S7-1200 CPU (ab Firmware 3.0) gesteckt werden.

Um die neuen Module zu verwenden, müssen Sie STEP 7 ab V11 SP2 Update 3 (Basic oder Professional) verwenden und Sie müssen das Hardware-Support-Paket (HSP) für neue Module über den Menübefehl **Optionen > Support-Pakete** in STEP 7 herunterladen. Befolgen Sie die Anweisungen zum Hinzufügen von Modulen zum Hardwarekatalog für STEP 7 (TIA-Portal), die Sie auf der Siemens-Website "Service & Support" (<http://support.automation.siemens.com>) finden.

## 1.5 Grundlegende HMI-Panels

Visualisierung gehört heute bei den meisten Maschinen zum Standardrepertoire. Deshalb bieten die SIMATIC HMI Basic Panels Geräte mit Touchscreen für grundlegende Aufgaben des Bedienens und Beobachtens. Alle Panels weisen die Schutzklasse IP65 auf und sind nach CE, UL, cULus und NEMA 4x zertifiziert.

Grundlegendes HMI-Panel	Beschreibung	Technische Daten
 <p>KP 300 Basic PN</p>	<p>3,6"-Membrantastatur mit 10 frei konfigurierbaren taktile Tasten</p> <ul style="list-style-type: none"> <li>• Mono (STN, schwarz/weiß)</li> <li>• 87 mm x 31 mm (3,6")</li> <li>• Programmierte Hintergrundfarbe (weiß, grün, gelb oder rot)</li> <li>• Auflösung: 240 x 80</li> </ul>	<ul style="list-style-type: none"> <li>• 250 Variablen</li> <li>• 50 Prozessbilder</li> <li>• 200 Meldungen</li> <li>• 25 Kurven</li> <li>• 40-KB-Rezeptspeicher</li> <li>• 5 Rezepte, 20 Datensätze, 20 Einträge</li> </ul>
 <p>KTP 400 Basic PN</p>	<p>4"-Touchscreen mit 4 taktile Tasten</p> <ul style="list-style-type: none"> <li>• Mono (STN, Graustufen)</li> <li>• 76,79 mm x 57,59 mm (3,8") Hoch- oder Querformat</li> <li>• Auflösung: 320 x 240</li> </ul>	<ul style="list-style-type: none"> <li>• 250 Variablen</li> <li>• 50 Prozessbilder</li> <li>• 200 Meldungen</li> <li>• 25 Kurven</li> <li>• 40-KB-Rezeptspeicher</li> <li>• 5 Rezepte, 20 Datensätze, 20 Einträge</li> </ul>
 <p>KTP 600 Basic PN</p>	<p>6"-Touchscreen mit 6 taktile Tasten</p> <ul style="list-style-type: none"> <li>• Farbe (TFT, 256 Farben) oder Mono (STN, Graustufen)</li> <li>• 115,2 mm x 86,4 mm (5,7") Hoch- oder Querformat</li> <li>• Auflösung: 320 x 240</li> </ul>	<ul style="list-style-type: none"> <li>• 500 Variablen</li> <li>• 50 Prozessbilder</li> <li>• 200 Meldungen</li> <li>• 25 Kurven</li> <li>• 40-KB-Rezeptspeicher</li> <li>• 5 Rezepte, 20 Datensätze, 20 Einträge</li> </ul>

Grundlegendes HMI-Panel	Beschreibung	Technische Daten
 <p style="text-align: center;">KTP 1000 Basic PN</p>	<p>10"-Touchscreen mit 8 taktilem Tasten</p> <ul style="list-style-type: none"> <li>• Farbe (TFT, 256 Farben)</li> <li>• 211,2 mm x 158,4 mm (10,4")</li> <li>• Auflösung: 640 x 480</li> </ul>	<ul style="list-style-type: none"> <li>• 500 Variablen</li> <li>• 50 Prozessbilder</li> <li>• 200 Meldungen</li> <li>• 25 Kurven</li> <li>• 40-KB-Rezeptspeicher</li> <li>• 5 Rezepte, 20 Datensätze, 20 Einträge</li> </ul>
 <p style="text-align: center;">TP 1500 Basic PN</p>	<p>15"-Touchscreen</p> <ul style="list-style-type: none"> <li>• Farbe (TFT, 256 Farben)</li> <li>• 304,1 mm x 228,1 mm (15,1")</li> <li>• Auflösung: 1024 x 768</li> </ul>	<ul style="list-style-type: none"> <li>• 500 Variablen</li> <li>• 50 Prozessbilder</li> <li>• 200 Meldungen</li> <li>• 25 Kurven</li> <li>• 40-KB-Rezeptspeicher (integrierter Flash)</li> <li>• 5 Rezepte, 20 Datensätze, 20 Einträge</li> </ul>



## STEP 7 Programmiersoftware

STEP 7 bietet eine bedienerfreundliche Umgebung zum Entwickeln, Bearbeiten und Beobachten der Logik zur Steuerung Ihrer Anwendung. Sie bietet auch die Werkzeuge zum Konfigurieren aller Geräte in Ihrem Projekt, wie PLC- und HMI-Geräte. Damit Sie die Informationen finden, die Sie benötigen, verfügt STEP 7 über eine umfangreiche Online-Hilfe.

STEP 7 bietet Standardprogrammiersprachen, mit denen Sie das Steuerungsprogramm für Ihre Anwendung bequem und effizient entwickeln können.

- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Darstellung beruht auf Schaltplänen (Seite 165).
- FUP (Funktionsplan) ist eine Programmiersprache, die auf den grafischen Logiksymbolen der Booleschen Algebra (Seite 166) basiert.
- SCL (Structured Control Language) ist eine textbasierte, höhere Programmiersprache.

Wenn Sie einen Codebaustein anlegen, müssen Sie die Programmiersprache für den Baustein auswählen. Ihr Anwenderprogramm kann mit Codebausteinen arbeiten, die in einer dieser Programmiersprachen angelegt wurden.

---

### Hinweis

STEP 7 ist die Softwarekomponente für Programmierung und Konfiguration im TIA-Portal. Das TIA-Portal umfasst neben STEP 7 auch WinCC zum Entwerfen und Ausführen von Runtime-Prozessvisualisierung, und es bietet Online-Hilfe für WinCC ebenso wie für STEP 7.

---

## 2.1 Systemvoraussetzungen

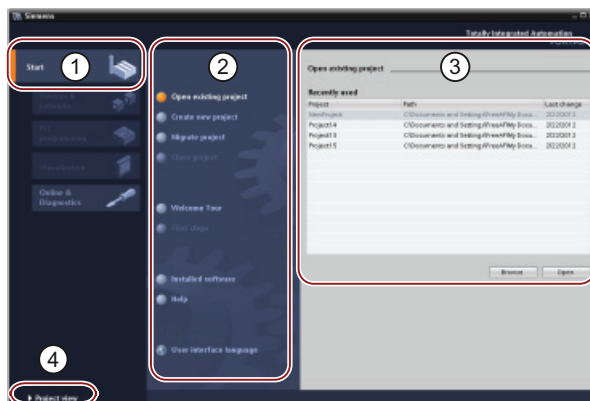
Zum Installieren der Software STEP 7 auf einem PC mit einem der Betriebssysteme Windows XP oder Windows 7 müssen Sie sich mit Administratorrechten anmelden.

Tabelle 2- 1 Systemvoraussetzungen

Hardware/Software	Voraussetzungen
Prozessortyp	Pentium M, 1,6 GHz oder ähnlich
RAM	1 GB
Freier Festplattenspeicher	2 GB auf Systemlaufwerk C:\
Betriebssysteme	<ul style="list-style-type: none"> <li>• Windows XP Professional SP3</li> <li>• Windows 2003 Server R2 StdE SP2</li> <li>• Windows 7 Home Premium (nur STEP 7 Basic, wird für STEP 7 Professional nicht unterstützt)</li> <li>• Windows 7 (Professional, Enterprise, Ultimate)</li> <li>• Windows 2008 Server StdE R2</li> </ul>
Grafikkarte	32 MB RAM-Speicher 24-Bit-Farbtiefe
Bildschirmauflösung	1024 x 768
Netzwerk	20 MBit/s Ethernet oder schneller
Optisches Laufwerk	DVD-ROM

## 2.2 Einfaches Arbeiten mit unterschiedlichen Ansichten

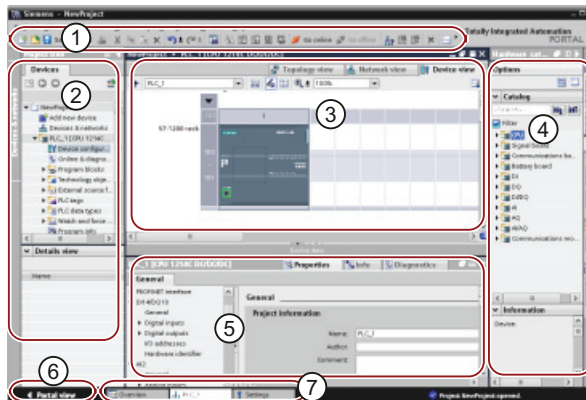
STEP 7 stellt eine benutzerfreundliche Umgebung bereit, in der Sie die Steuerungslogik entwickeln, die HMI-Visualisierung konfigurieren und die Netzwerkkommunikation einrichten können. Zur Steigerung Ihrer Produktivität bietet STEP 7 zwei unterschiedliche Ansichten des Projekts: eine tätigkeitsorientierte Anzahl von Portalen für die einzelnen Funktionen (Portalansicht) und eine projektorientierte Ansicht der Elemente im Projekt (Projektansicht). Sie entscheiden, in welcher Ansicht Sie am effizientesten arbeiten können. Per Mausclick können Sie zwischen der Portalansicht und der Projektansicht wechseln.



### Portalansicht

- ① Portale für die verschiedenen Aufgaben
- ② Aufgaben für das ausgewählte Portal
- ③ Auswahlpanel zur gewählten Aktion
- ④ Änderungen an der Projektansicht





#### Projektansicht

- ① Menüs und Funktionsleiste
- ② Projektnavigator
- ③ Arbeitsbereich
- ④ Taskcards
- ⑤ Inspektorfenster
- ⑥ Änderungen an der Portalansicht
- ⑦ Editorleiste

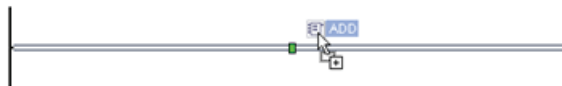
Da sich alle Komponenten an einer Stelle befinden, haben Sie schnellen Zugriff auf jeden Bereich Ihres Projekts. Beispielsweise zeigt das Inspektorfenster die Eigenschaften und weitere Informationen für das Objekt, das im Arbeitsbereich ausgewählt wurde. Für die verschiedenen von Ihnen gewählten Objekte zeigt das Inspektorfenster jeweils die konfigurierbaren Eigenschaften. Das Inspektorfenster verfügt außerdem über Register, unter denen Diagnoseinformation und weitere Meldungen angezeigt werden.

In der Editorleiste werden alle derzeit geöffneten Editoren angezeigt. Mit der Editorleiste arbeiten Sie so schneller und effizienter. Zum Umschalten zwischen geöffneten Editoren klicken Sie einfach auf den gewünschten Editor. Sie können auch zwei Editoren gleichzeitig anzeigen und diese vertikal oder horizontal anordnen. Dadurch sind "Drag&Drop"-Operationen zwischen Editoren möglich.

## 2.3 Bedienerfreundliche Werkzeuge

### 2.3.1 Anweisungen in Ihr Anwenderprogramm einfügen

STEP 7 bietet Taskcards mit den Anweisungen für Ihr Programm. Die Anweisungen sind nach Funktionen gegliedert.

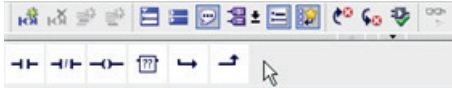


Um Ihr Programm anzulegen, ziehen Sie die Anweisungen von der Taskcard in ein Netzwerk.

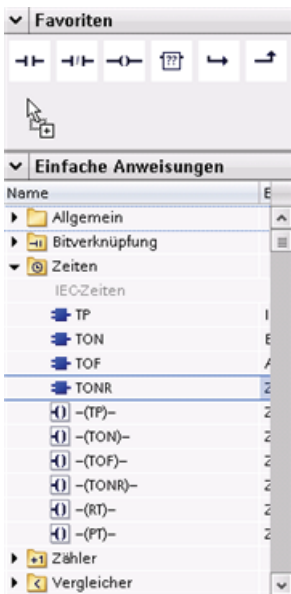
Einfache Anweisungen	
Name	
▶	Allgemein
▶	Bitverknüpfung
▶	Zeiten
▶	Zähler
▶	Vergleicher
▶	Mathematische Funktionen
▶	Verschieben
▶	Umwandler
▶	Programmsteuerung
▶	Wortverknüpfungen
▶	Schieben und Rotieren

### 2.3.2 Zugreifen auf Anweisungen in der Funktionsleiste "Favoriten"

STEP 7 enthält eine Funktionsleiste "Favoriten" für den schnellen Zugriff auf Anweisungen, die Sie häufig verwenden. Um eine Anweisung in Ihr Netzwerk einzufügen, genügt ein Klick auf das entsprechende Symbol.



(Um im Anweisungsverzeichnis die "Favoriten" aufzurufen, doppelklicken Sie auf das Symbol.)



Die Funktionsleiste "Favoriten" lässt sich problemlos anpassen und durch neue Anweisungen erweitern.

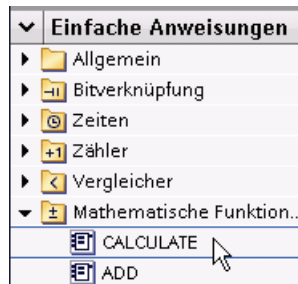
Fügen Sie die gewünschte Anweisung einfach mit "Drag & Drop" den "Favoriten" hinzu.

Für den Zugriff auf die Anweisung genügt jetzt ein Klick!

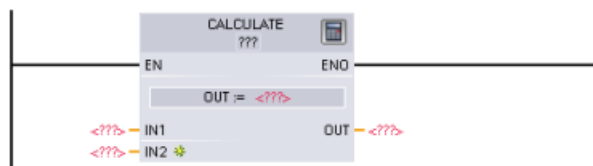


### 2.3.3 Erstellen einer komplexen Gleichung mit einer einfachen Anweisung

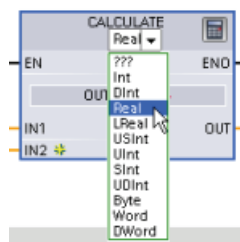
Mit der Anweisung Calculate können Sie eine mathematische Funktion erstellen, die mehrere Eingangsparameter verarbeitet und das Ergebnis entsprechend der von Ihnen vorgegebenen Gleichung ausgibt.



Erweitern Sie im Basic-Anweisungsverzeichnis den Ordner der mathematischen Funktionen. Doppelklicken Sie auf die Anweisung Calculate, um sie in Ihr Anwenderprogramm einzufügen.



Die nicht konfigurierte Anweisung Calculate bietet zwei Eingangsparameter und einen Ausgangsparameter.



Klicken Sie auf "???" und wählen Sie die Datentypen für die Eingangs- und Ausgangsparameter aus. (Alle Eingangs- und Ausgangsparameter müssen denselben Datentyp haben.)

Wählen Sie für dieses Beispiel den Datentyp "Real" aus.



Klicken Sie auf das Symbol "Gleichung bearbeiten", um die Gleichung einzugeben.



Geben Sie in diesem Beispiel die folgende Gleichung zum Skalieren eines Rohanalogwerts ein. (Die Bezeichnungen "In" und "Out" entsprechen den Parametern der Anweisung Calculate.)

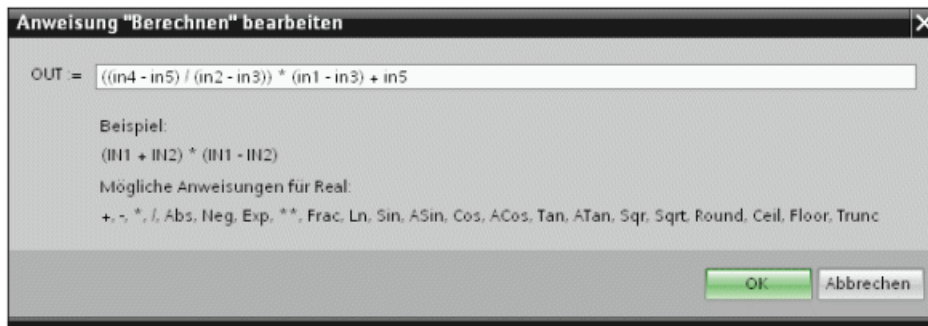
$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) * (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

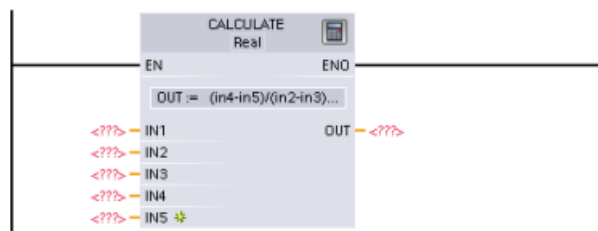
Erläuterung:	Out <sub>value</sub>	(Out)	Skalierter Ausgangswert
	In <sub>value</sub>	(in1)	Analogeingangswert
	In <sub>high</sub>	(in2)	Oberer Grenzwert für den skalierten Eingangswert
	In <sub>low</sub>	(in3)	Unterer Grenzwert für den skalierten Eingangswert
	Out <sub>high</sub>	(in4)	Oberer Grenzwert für den skalierten Ausgangswert
	Out <sub>low</sub>	(in5)	Unterer Grenzwert für den skalierten Ausgangswert

Geben Sie im Feld "Calculate bearbeiten" die Gleichung mit den Parameternamen ein:

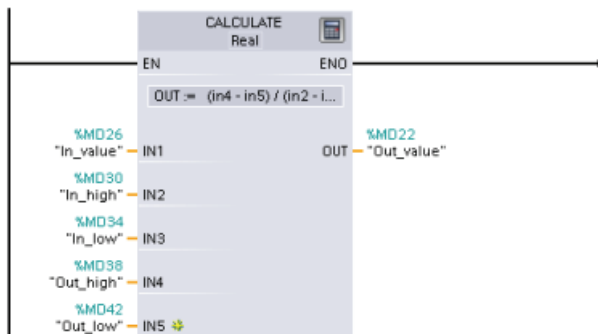
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$



Wenn Sie auf "OK" klicken, erstellt die Anweisung Calculate die für die Anweisung erforderlichen Eingänge.



Geben Sie die Variablennamen für die Werte ein, die den Parametern entsprechen.



## 2.3.4 Ergänzen von Eingängen oder Ausgängen in einer KOP- oder FUP-Anweisung

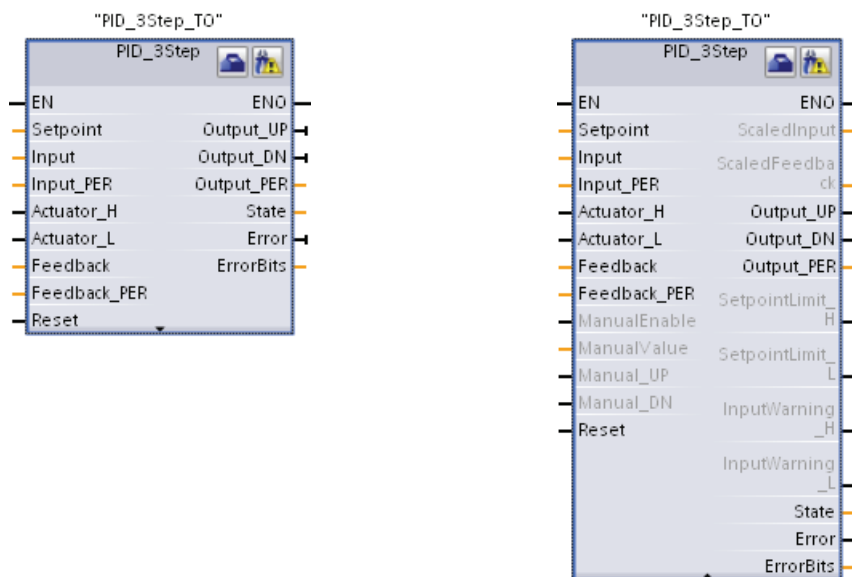


Bei einigen Anweisungen können Sie weitere Eingänge oder Ausgänge erstellen.

- Um einen Eingang oder Ausgang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN oder OUT mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".
- Um einen Eingang oder Ausgang zu löschen, klicken Sie bei einem der vorhandenen IN- oder OUT-Parameter mit der rechten Maustaste auf den Anschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

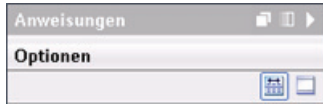
## 2.3.5 Erweiterbare Anweisungen

Einige der komplexeren Anweisungen sind erweiterbar und zeigen zunächst nur die wesentlichen Eingänge und Ausgänge an. Um die Eingänge und Ausgänge vollständig anzuzeigen, klicken Sie auf den Pfeil im unteren Bereich der Anweisung.

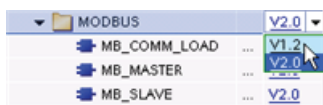


### 2.3.6 Auswählen einer Version für eine Anweisung

Durch die Entwicklung und die Ausgabezyklen bestimmter Befehlssätze (z. B. Modbus, PID und Bewegungssteuerung) ist es inzwischen zu mehreren freigegebenen Versionen dieser Anweisungen gekommen. Um die Kompatibilität und Migration mit älteren Projekten sicherzustellen, können Sie in STEP 7 auswählen, welche Anweisungsversion Sie in Ihr Anwenderprogramm einfügen.

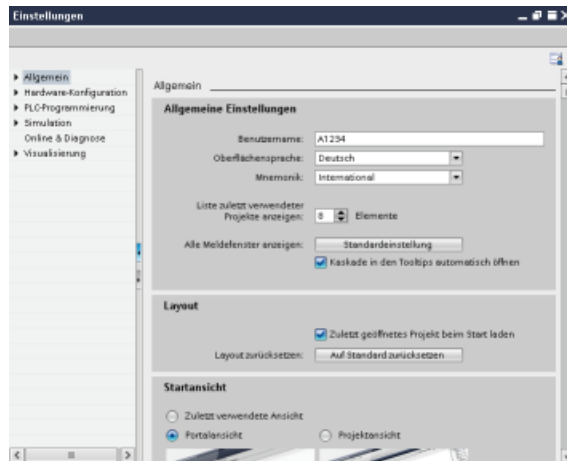


Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.



Um die Version einer Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus.

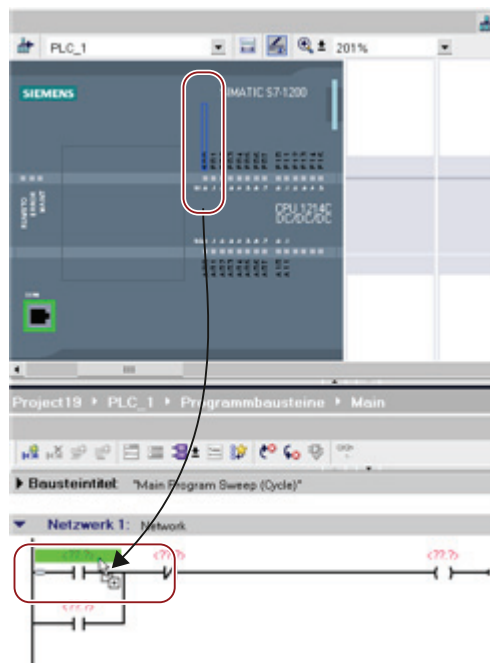
## 2.3.7 Ändern des Erscheinungsbilds und der Konfiguration von STEP 7



Sie haben zahlreiche Einstellmöglichkeiten; diese betreffen z. B. das Aussehen der Bedienoberfläche, die Sprache oder den Ordner zum Speichern Ihrer Arbeitsergebnisse.

Wählen Sie zum Ändern von Einstellungen im Menü "Optionen" den Befehl "Einstellungen".

## 2.3.8 Drag & Drop zwischen Editoren



Damit Sie Aufgaben schnell und unkompliziert erledigen können, ermöglicht STEP 7 das Ziehen und Ablegen mit der Maus ("Drag & Drop") von Elementen zwischen den Editoren. So können Sie beispielsweise einen Eingang von der CPU an die Adresse einer Anweisung in Ihrem Anwenderprogramm ziehen.

Sie brauchen zum Auswählen der Eingänge oder Ausgänge der CPU mindestens den Zoomfaktor 200 %.

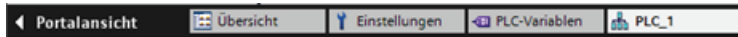
Beachten Sie, dass die Variablenamen nicht nur in der PLC-Variablen-tabelle, sondern auch in der CPU angezeigt werden.

Um zwei Editoren gleichzeitig anzuzeigen, verwenden Sie den Menübefehl "Editor teilen" oder die entsprechende Schaltfläche in der Funktionsleiste.

Fenster	Hilfe
Alle schließen	Strg+Shift+F4
Alle minimieren	
Nächster Editor	Strg+F6
Vorheriger Editor	Strg+Shift+F6
Editorbereich vertikal teilen	
Editorbereich horizontal teilen	Shift+F9



Zum Umschalten zwischen den geöffneten Editoren klicken Sie auf die jeweiligen Symbole in der Editorleiste.



### 2.3.9 Wechseln des Betriebszustands der CPU

Siehe

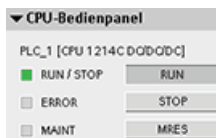
Die CPU verfügt nicht über einen physischen Schalter zum Ändern des Betriebszustands (STOP oder RUN).

Klicken Sie in der Funktionsleiste auf die Schaltfläche "CPU starten" oder "CPU stoppen", um den Betriebszustand der CPU zu ändern.



Beim Konfigurieren der CPU in der Gerätekonfiguration legen Sie das Anlaufverhalten der CPU über ihre Eigenschaften fest.

Im Portal "Online & Diagnose" steht Ihnen auch ein Bedienpanel zur Verfügung, in dem Sie den Betriebszustand der Online-CPU ändern können. Um das CPU-Bedienpanel verwenden zu können, müssen Sie mit der CPU online verbunden sein. Das in der Taskcard "Online-Tools" enthaltene Bedienpanel zeigt den Betriebszustand der Online-CPU an. Über das Bedienpanel können Sie auch den Betriebszustand der Online-CPU ändern.



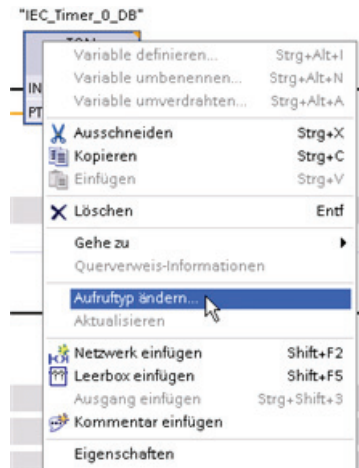
Mit der Schaltfläche auf dem Bedienpanel ändern Sie den Betriebszustand (STOP bzw. RUN). Außerdem enthält das Bedienpanel eine Schaltfläche MRES zum Umräumen des Speichers.

Der aktuelle Betriebszustand der CPU wird durch die Farbe der RUN/STOP-Anzeige angegeben. Gelb steht für den Betriebszustand STOP, Grün für RUN.

Unter Betriebszustände der CPU im S7-1200 Systemhandbuch (Seite 75) finden Sie weitere Informationen zum Konfigurieren des standardmäßigen Betriebszustands beim Hochlauf.



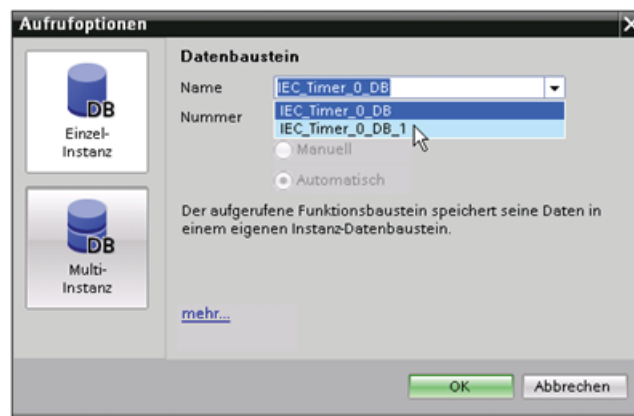
### 2.3.10 Aufruftyp eines DB ändern



In STEP 7 können Sie problemlos die Zuweisung einrichten oder ändern, die zwischen einem DB und einer Anweisung oder einem DB und einem FB besteht, der sich in einem FB befindet.

- Sie können die Zuweisung zwischen verschiedenen DBs umschalten.
- Sie können die Zuweisung zwischen einem Einzelinstanz-DB und einem Multiinstanz-DB wechseln.
- Sie können einen Instanz-DB erstellen (sofern ein Instanz-DB fehlt oder nicht verfügbar ist).

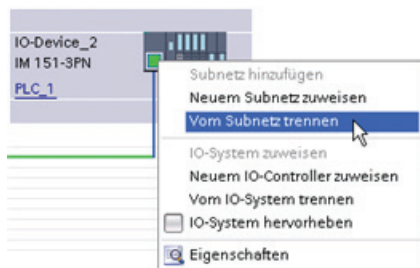
Sie können den Befehl "Aufruftyp ändern" auswählen, indem Sie entweder im Programmiereditor mit der rechten Maustaste auf die Anweisung oder den FB klicken oder indem Sie im Menü "Optionen" den Befehl "Bausteinaufruf" auswählen.



Im Dialog "Aufrufoptionen" können Sie einen Einzelinstanz- oder einen Multiinstanz-DB auswählen. Außerdem können Sie in einer Klappliste der verfügbaren DBs bestimmte DBs auswählen.

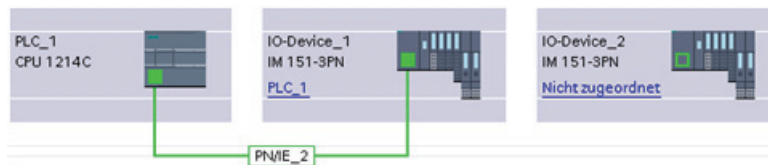
### 2.3.11 Geräte vorübergehend vom Netzwerk trennen

Sie können einzelne Netzwerkgeräte vom Subnetz trennen. Weil die Konfiguration des Geräts nicht aus dem Projekt entfernt wird, können Sie die Verbindung des Geräts mühelos wiederherstellen.



Klicken Sie mit der rechten Maustaste auf den Schnittstellenanschluss des Netzwerkgeräts und wählen Sie im Kontextmenü den Befehl "Vom Subnetz trennen".

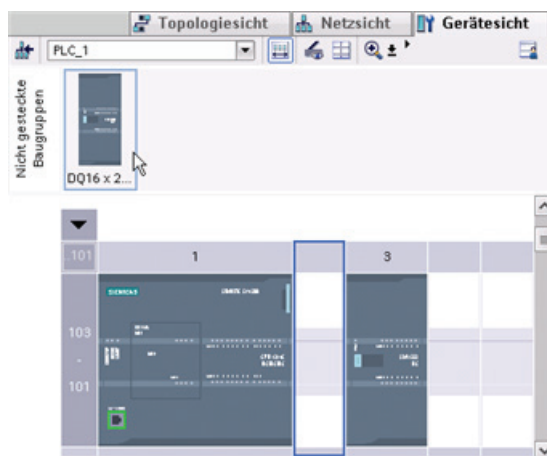
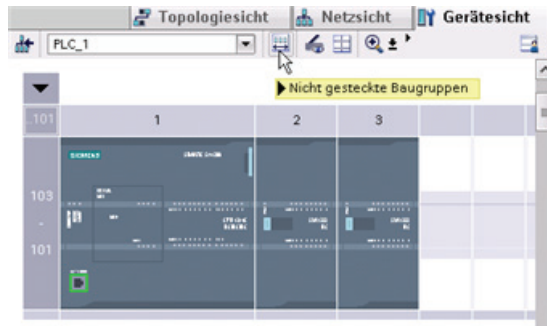
STEP 7 konfiguriert die Netzwerkverbindungen neu, entfernt das getrennte Gerät jedoch nicht aus dem Projekt. Die Netzwerkverbindung wird zwar gelöscht, doch die Schnittstellenadressen werden nicht verändert.



Wenn Sie die neuen Netzwerkverbindungen laden, müssen Sie die CPU in den Betriebszustand STOP versetzen.

Um das Gerät wieder anzuschließen, erstellen Sie einfach eine neue Netzwerkverbindung mit dem Anschluss des Geräts.

## 2.3.12 Virtuelles Abziehen von Geräten aus der Konfiguration



STEP 7 bietet Ihnen eine virtuelle Ablage für "nicht gesteckte" Baugruppen. Sie können eine Baugruppe vom Baugruppenträger "abziehen" und dabei die Konfiguration der Baugruppe speichern. Diese abgezogenen Baugruppen werden mit dem Projekt zusammen gespeichert; dadurch können Sie sie später wieder einsetzen, ohne die Parameter erneut konfigurieren zu müssen.

Diese Funktion wird z. B. bei kurzfristigen Wartungsarbeiten eingesetzt. Denken Sie beispielsweise an einen Fall, in dem Sie auf eine bestimmte Baugruppe warten, die ausgetauscht werden soll, und eine andere Baugruppe vorübergehend als Ersatz verwenden wollen. In diesem Fall können Sie die konfigurierte Baugruppe vom Baugruppenträger in die "Ablage nicht gesteckter Baugruppen" ziehen und anschließend die vorübergehende Ersatzbaugruppe einsetzen.



## 3.1 Richtlinien für den Einbau von S71200 Geräten

Die S7-1200 Geräte wurden so ausgelegt, dass sie einfach einzubauen sind. Sie können eine S7-1200 entweder in einer Schalttafel oder auf einer Standard-Hutschiene einbauen; die S71200 kann horizontal oder vertikal eingebaut werden. Die kompakte Größe der S7-1200 macht eine effiziente Platzausnutzung möglich.

 <b>WARNUNG</b>
--

<p>Bei den SIMATIC S7-1200 Automatisierungssystemen handelt es sich um offene Steuerungen. Sie müssen die S7-1200 in einem Gehäuse, Schaltschrank oder in einer Schaltzentrale einbauen. Nur berechtigtes Personal darf Zugang zum Gehäuse, Schaltschrank oder der Schaltzentrale haben.</p>
--

<p>Die Nichteinhaltung dieser Einbaubedingungen kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.</p>
---

<p>Beachten Sie diese Maßnahmen beim Einbau der S7-1200 Automatisierungssysteme.</p>
--

### Halten Sie die S7-1200 Geräte fern von Wärme, Hochspannung und elektrischen Störungen


Als allgemeine Regel für die Anordnung von Geräten in Ihrem System gilt, dass Sie Geräte, die Hochspannung oder hohe elektrische Störungen erzeugen, von den elektronischen Niederspannungsgeräten wie der S7-1200 fernhalten.

Wenn Sie das Layout der S71200 in Ihrer Schalttafel planen, berücksichtigen Sie wärmeerzeugende Geräte und ordnen Sie die elektronischen Geräte in den kühleren Bereichen Ihres Schaltschranks an. Je weniger Sie ein elektronisches Gerät Umgebungen mit hohen Temperaturen aussetzen, desto länger ist die Betriebsdauer des Geräts.

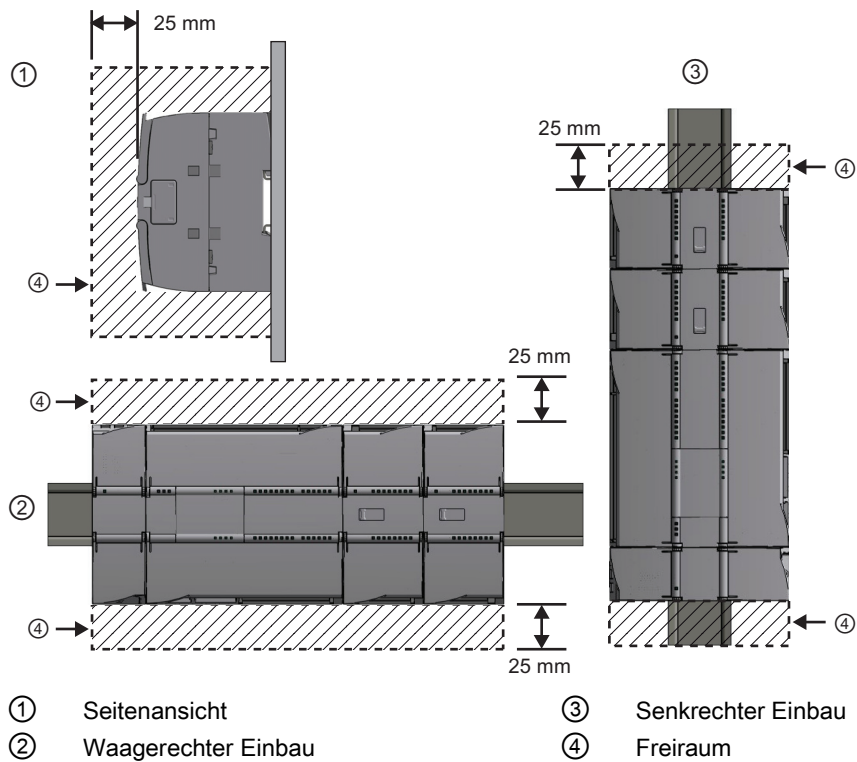
Berücksichtigen Sie auch, wie Sie die Verdrahtung der Geräte in der Schalttafel verlegen. Vermeiden Sie es, Niederspannungssignalleitungen und Kommunikationskabel in der gleichen Kabelbahn wie AC-Versorgungsleitungen und schnellschaltende Hochgeschwindigkeits-DC-Leitungen zu verlegen.

### Lassen Sie genügend Abstand für Kühlung und Verdrahtung

Die S71200 Geräte sind für natürliche Wärmeabfuhr durch Konvektion ausgelegt. Lassen Sie deshalb oberhalb und unterhalb der Geräte jeweils mindestens 25 mm Platz, um die Wärmeabfuhr zu gewährleisten. Achten Sie ferner darauf, dass zwischen der Modulfront und der Innenseite des Gehäuses eine Tiefe von mindestens 25 mm bleibt.

 <b>VORSICHT</b>
Bei vertikalem Einbau ist die maximal zulässige Umgebungstemperatur um 10 °C niedriger. Richten Sie ein vertikal eingebautes S71200 System wie in der folgenden Abbildung gezeigt aus.

Wenn Sie das Layout für Ihr S71200 System planen, lassen Sie genügend Abstand für die Verdrahtung und die Kommunikationskabelanschlüsse.



## 3.2 Leistungsbilanz

Ihre CPU besitzt eine interne Spannungsversorgung, die neben der CPU die Signalmodule, Signalboards, Kommunikationsmodule und andere 24VDC-Verbraucher speist.

In den Technischen Daten (Seite 743) finden Sie Informationen zur 5-V-DC-Leistungsbilanz Ihrer CPU und zum 5-V-DC-Leistungsbedarf der Signalmodule, Signalboards und Kommunikationsmodule. Mit Hilfe der Informationen unter "Berechnen der Leistungsbilanz" (Seite 879) können Sie berechnen, wieviel Leistung (bzw. Strom) die CPU für Ihre Konfiguration liefern kann.

Die CPU verfügt auch über eine 24-V-DC-Geberversorgung, die 24 V DC für Eingänge, für die Relaispulen der Signalmodule und für sonstige Verbraucher liefert. Wenn der 24-V-DC-Leistungsbedarf die Leistung der Geberversorgung übersteigt, dann müssen Sie eine externe 24-V-DC-Spannungsversorgung an Ihr System anschließen. In den Technischen Daten (Seite 743) finden Sie die Leistungsbilanz für die 24-V-DC-Geberversorgung für Ihre jeweilige S7-1200 CPU.

---

### Hinweis

Das CM 1243-5 (PROFIBUS-Mastermodul) muss über die 24-V-DC-Geberversorgung der CPU gespeist werden.

---

Wenn Sie eine externe 24-V-DC-Spannungsversorgung benötigen, müssen Sie darauf achten, dass die Spannungsversorgung nicht parallel zur Geberversorgung der CPU angeschlossen ist. Die beste Störfestigkeit erreichen Sie, wenn Sie die Leitungen (M) der jeweiligen Spannungsversorgungen miteinander verbinden.

 <b>WARNUNG</b>
--

Wenn Sie parallel zu der 24-DC-Geberversorgung eine externe 24VDC-Spannungsquelle anschließen, kann es sein, dass die beiden Spannungsquellen sich beim Aufbauen der geeigneten Ausgangsspannung gegenseitig beeinträchtigen.
---

Als Folge kann sich die Lebensdauer verkürzen bzw. eine oder beide Spannungsquellen können sofort ausfallen. Unvorhersehbarer Betrieb kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.
---

Die DC-Geberversorgung und eine externe Spannungsquelle müssen die Spannung an unterschiedlichen Punkten liefern.
---

Einige der 24-V-DC-Eingangsport des S7-1200 Systems sind miteinander verbunden, wobei ein logischer Bezugsleiter mehrere M-Klemmen verbindet. Die folgenden Stromkreise sind beispielsweise miteinander verbunden, sofern sie in den Datenblättern als "nicht potentialgetrennt" angegeben sind: die 24-V-DC-Versorgung der CPU, der Leistungseingang für die Relaispule eines SM oder die Versorgung eines nicht potentialgetrennten Analogeingangs. Alle nicht potentialgetrennten M-Klemmen müssen an dasselbe externe Bezugspotential angeschlossen werden.

 **WARNUNG**

Wenn Sie nicht potentialgetrennte M-Klemmen an verschiedene Bezugspotentiale anschließen, verursacht dies unbeabsichtigten Stromfluss, der zu Beschädigung oder unvorhersehbarem Betrieb des Zielsystems und angeschlossener Geräte führen kann.

Die Nichteinhaltung dieser Richtlinien kann Schaden oder unvorhersehbaren Betrieb verursachen, was zu Tod oder schwerer Körperverletzung und/oder Sachschaden führen kann.

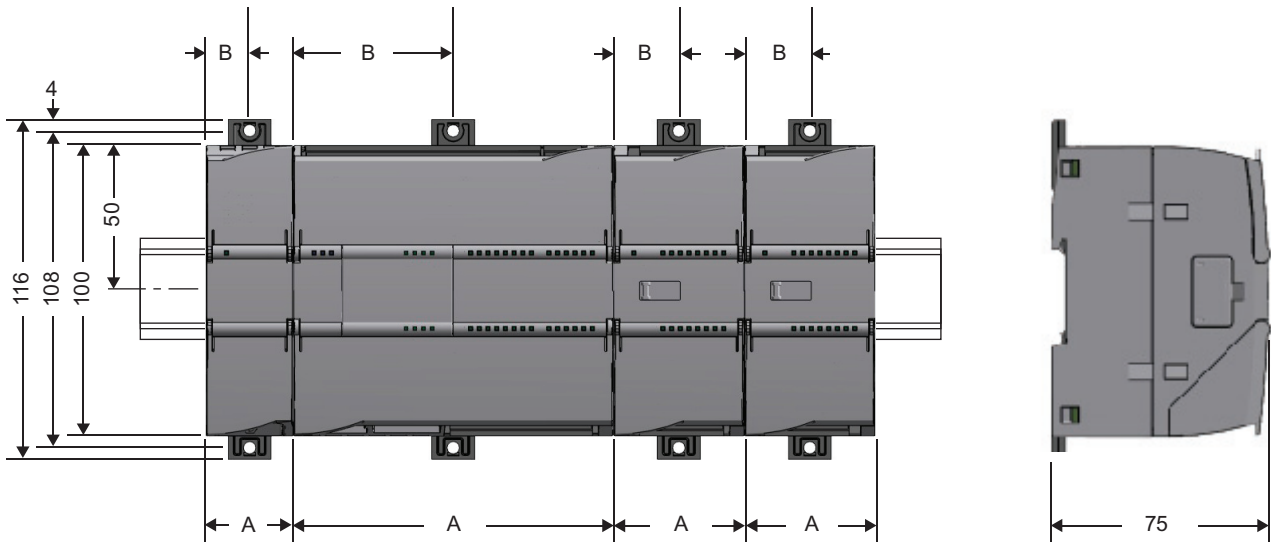
Schließen Sie stets alle nicht potentialgetrennten M-Klemmen in einem S7-1200 System an dasselbe Bezugspotential an.



### 3.3 Ein- und Ausbauanweisungen

#### 3.3.1 Einbaumaße der S7-1200 Geräte

CPU 1211C, CPU 1212C, CPU 1214C



CPU 1215C

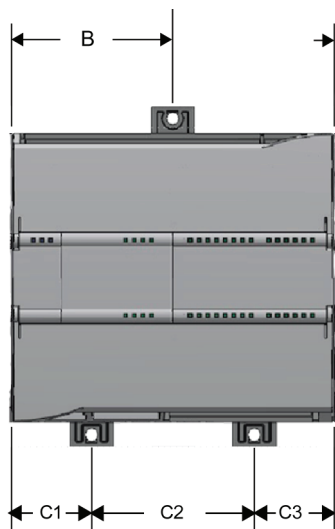


Tabelle 3- 1 Abmessungen für die Montage (mm)

S71200 Geräte		Breite A (mm)	Breite B (mm)	Breite C (mm)
CPU	CPU 1211C und CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (oben)	Unten: C1: 32.5 C2: 65 C3: 32.5
Signalmodule	8 oder 16 digitale E/A 2, 4 oder 8 analoge E/A Thermoelement, 4 oder 8 E/A RTD, 4 E/A	45	22.5	--
	8 digitale Ausgänge x Relais (Umschaltung)	70	22.5	--
	16 analoge E/A RTD, 8 E/A	70	35	--
Kommunikationsschnittstellen	CM 1241 RS232 und CM 1241 RS422/485 CM 1243-5 PROFIBUS-Master und CM 1242-5 PROFIBUS-Slave CM 1242-2 AS-i-Master CP 1242-7 GPRS	30	15	--
	TS-Adapter IE Basic	60 <sup>1</sup>	15	--

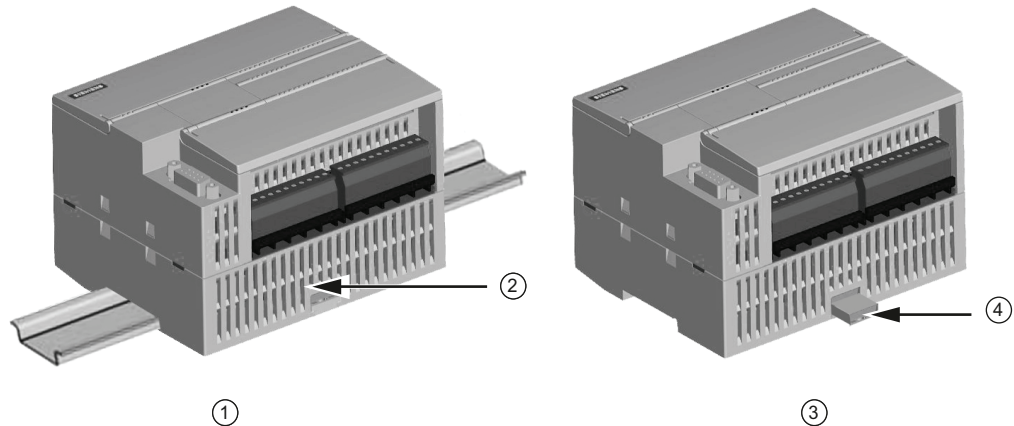
<sup>1</sup> Weil Sie zusammen mit dem TS-Adapter ein TS-Adaptermodul einbauen müssen, beträgt die Gesamtbreite ("Breite A") 60 mm.

Alle CPUs, SMs, CMs und CPs können auf der DIN-Schiene oder im Schaltschrank montiert werden. Verwenden Sie die Hutschienenklammern für die Befestigung des Geräts auf der Hutschiene. Diese Klammern rasten auch in einer ausgezogenen Position ein, um den Einbau des Geräts in einer Schalttafel zu ermöglichen. Das Innenmaß der Bohrung für die Hutschienenklammern am Gerät beträgt 4,3 mm.

Ober- und unterhalb des Geräts muss ein Freiraum von 25 mm für die Luftzirkulation als Schutz vor Überhitzung eingehalten werden.

## Einbau und Ausbau der S71200 Geräte

Die CPU kann auf einfache Weise auf einer Standard-Hutschiene oder in einer Schalttafel eingebaut werden. Für die Befestigung des Geräts auf der Hutschiene werden passende Hutschienenklemmen mitgeliefert. Diese Klemmen können auch in einer ausgezogenen Position einrasten und ermöglichen dann das Festschrauben des Geräts in einer Schalttafel.



- |   |  |
|---|--|
| ① Hutschienenmontage                                  | ③ Schalttafelmontage   |
| ② Standard-Hutschienenklemme in verriegelter Position | ④ Klemme in ausgezogener Position für die Schalttafelmontage |

Bevor Sie ein elektrisches Gerät einbauen oder ausbauen, müssen Sie sicherstellen, dass die Spannungsversorgung der Geräte ausgeschaltet ist. Achten Sie außerdem darauf, dass auch alle angeschlossenen Geräte ausgeschaltet sind.

### **WARNUNG**

Wenn Sie die S7-1200 oder daran angeschlossene Geräte in eingeschaltetem Zustand ein- oder ausbauen, kann es passieren, dass Sie einen elektrischen Schlag bekommen oder die Geräte unerwartet arbeiten.

Ist die Spannungsversorgung der S7-1200 und aller daran angeschlossenen Geräte während des Einbaus bzw. Ausbaus von Geräten nicht abgeschaltet, so kann dies aufgrund von elektrischem Schlag oder unerwartetem Betrieb der Geräte zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, dass vor dem Einbau bzw. Ausbau eines Geräts die Spannungsversorgung der S71200 CPUs abgeschaltet ist.

Achten Sie immer darauf, dass Sie das richtige Modul bzw. das richtige Gerät verwenden, wenn Sie ein S7-1200 Gerät einbauen bzw. austauschen.

 **WARNUNG**

Falscher Einbau eines S7-1200 Moduls kann zu unvorhersehbarer Funktionsweise des Programms der S7-1200 führen.

Wird ein S7-1200 Gerät durch eine andere Variante ersetzt, nicht richtig ausgerichtet oder in der falschen Reihenfolge eingebaut, so kann dies aufgrund von unerwartetem Betrieb der Geräte zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Wechseln Sie ein S7-1200 Gerät immer mit der gleichen Ausführung aus, richten Sie das Gerät korrekt aus und bauen Sie es an der richtigen Stelle ein.

 **WARNUNG**

Trennen Sie keine Geräte, wenn eine entflammbare oder brennbare Atmosphäre vorliegt.

Das Trennen von Geräten in einer entflammbaren oder brennbaren Atmosphäre kann ein Feuer oder eine Explosion verursachen, was zu Tod, schweren Verletzungen und/oder Sachschaden führen kann.

**VORSICHT**

Elektrostatische Entladungen können das Gerät oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit dem Gerät arbeiten.

### 3.3.2 Einbau und Ausbau der CPU

Sie können die CPU in einer Schalttafel oder auf einer Standard-Hutschiene einbauen.

---

**Hinweis**

Schließen Sie die Kommunikationsmodule an die CPU an und bauen Sie alle Module gemeinsam ein. Installieren Sie die Signalmodule erst nach dem Einbau der CPU.

---

Beachten Sie beim Einbau der Geräte auf der Hutschiene oder in einer Schalttafel Folgendes:

- Achten Sie beim Einbau in eine Hutschiene darauf, dass sich die obere Hutschienenklemme der CPU und der angeschlossenen CMs in der eingerasteten (inneren) Stellung und die untere Hutschienenklemme in der ausgezogenen Stellung befindet.
- Bringen Sie nach dem Einbau der Geräte auf der Hutschiene die unteren Hutschienenklemmen in die eingerastete Stellung, um die Geräte auf der Hutschiene zu befestigen.
- Achten Sie bei der Schalttafelmontage darauf, dass sich die Hutschienenklemmen in der ausgezogenen Stellung befinden.

Um die CPU in eine Schalttafel einzubauen, gehen Sie folgendermaßen vor:

1. Bringen Sie die Bohrungen (M4) mit den in den Abmessungen in der Tabelle Abmessungen für die Montage (mm) (Seite 49) gezeigten Vorgaben an.
2. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.
3. Ziehen Sie die Montageklemmen aus dem Modul heraus. Achten Sie darauf, dass sich die Hutschienenklemmen oben und unten an der CPU in der ausgezogenen Position befinden.
4. Schrauben Sie das Modul mit einer Zylinderkopfschraube M4 sowie einem Federring und einer flachen Unterlegscheibe fest. Verwenden Sie keine Senkkopfschraube.

---

**Hinweis**

Welcher Schraubentyp erforderlich ist, hängt von der Art des Materials ab, auf dem das Modul montiert wird. Ziehen Sie die Schraube mit dem entsprechenden Drehmoment fest, bis der Federring flachgedrückt ist. Ziehen Sie die Schrauben nicht mit übermäßigem Drehmoment fest. Verwenden Sie keine Senkkopfschrauben.

---

**Hinweis**

Ist Ihr System starken Schwingungen ausgesetzt oder vertikal montiert, bietet die Schalttafelmontage der S7-1200 einen besseren Schutz.

---

3.3 Ein- und Ausbauanweisungen

Tabelle 3-2 Einbau der CPU auf einer DIN-Schiene

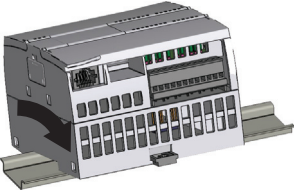
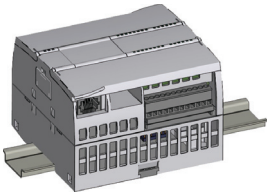
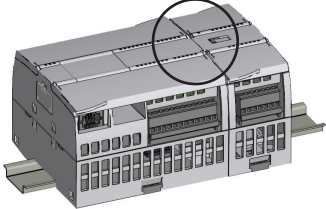
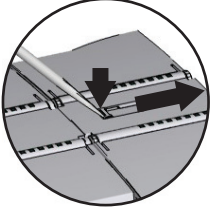
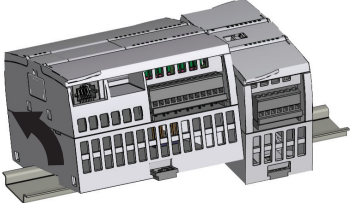
Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> <li>1. Montieren Sie die Hutschiene. Verschrauben Sie die Hutschiene in Abständen von jeweils 75 mm mit der Schalttafel.</li> <li>2. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>3. Hängen Sie die CPU oben an der Hutschiene ein.</li> <li>4. Ziehen Sie die Hutschieneklemme auf der Unterseite der CPU heraus, damit die CPU über die Schiene passt.</li> </ol>
	<ol style="list-style-type: none"> <li>5. Schwenken Sie die CPU nach unten in ihre Einbauposition auf der Schiene.</li> <li>6. Drücken Sie die Klemmen herunter, so dass die CPU fest auf der Schiene sitzt.</li> </ol>

Tabelle 3-3 Ausbau der CPU auf einer DIN-Schiene

Aufgabenstellung	Vorgehensweise	
		<ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Trennen Sie die E/A-Steckverbinder, die Verdrahtung und Kabel von der CPU (Seite 60).</li> <li>3. Bauen Sie die CPU und die zugehörigen Kommunikationsmodule gemeinsam aus. Alle Signalmodule müssen eingebaut bleiben.</li> </ol>
	<ol style="list-style-type: none"> <li>4. Wenn ein SM an die CPU angeschlossen ist, ziehen Sie den Busstecker zurück: <ul style="list-style-type: none"> <li>– Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des Signalmoduls an.</li> <li>– Drücken Sie nach unten, um den Klemmenblock von der CPU zu lösen.</li> <li>– Schieben Sie die Lasche ganz nach rechts.</li> </ul> </li> <li>5. Nehmen Sie die CPU ab: <ul style="list-style-type: none"> <li>– Ziehen Sie die Hutschieneklemme heraus, damit die CPU von der Schiene gelöst wird.</li> <li>– Schwenken Sie die CPU nach oben von der Schiene herunter und nehmen Sie die CPU aus dem System heraus.</li> </ul> </li> </ol>	

### 3.3.3 Ein- und Ausbau eines SBs, CBs oder BBs

Tabelle 3- 4 Einbau eines SBs, CBs oder BBs 1297

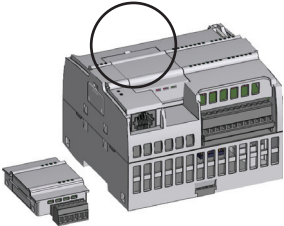
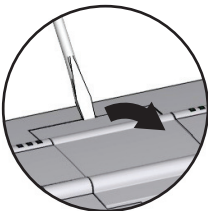
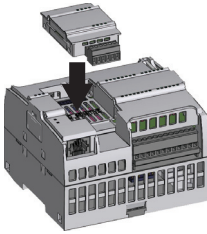

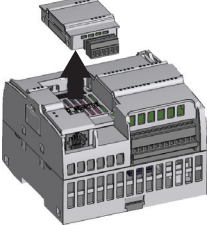
Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Nehmen Sie die obere und untere Abdeckung des Klemmenblocks von der CPU ab.</li> <li>3. Führen Sie einen Schraubendreher in den Schlitz oben auf der CPU an der Hinterseite der Abdeckung ein.</li> <li>4. Hebeln Sie die Abdeckung vorsichtig aus ihrer Halterung und nehmen Sie sie von der CPU ab.</li> </ol>
 	<ol style="list-style-type: none"> <li>5. Setzen Sie das Modul gerade von oben in seine Montageposition in der CPU ein.</li> <li>6. Drücken Sie das Modul fest herunter, bis es einrastet.</li> <li>7. Setzen Sie die Klemmenabdeckungen wieder ein.</li> </ol>

Tabelle 3- 5 Ausbau eines SBs, CBs oder BBs 1297

Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Nehmen Sie die obere und untere Abdeckung des Klemmenblocks von der CPU ab.</li> <li>3. Setzen Sie einen Schraubendreher in den Schlitz auf der Oberseite des Moduls ein.</li> <li>4. Hebeln Sie das Modul vorsichtig aus seiner Halterung und nehmen Sie es von der CPU ab.</li> </ol>
	<ol style="list-style-type: none"> <li>5. Nehmen Sie das Modul gerade von oben aus seiner Montageposition aus der CPU heraus.</li> <li>6. Bringen Sie die Abdeckung der CPU wieder an.</li> <li>7. Setzen Sie die Klemmenabdeckungen wieder ein.</li> </ol>

### Einsetzen oder Austauschen der Batterie des Batterieboards BB 1297

Das BB 1297 benötigt eine Batterie des Typs CR1025. Die Batterie ist nicht im Lieferumfang des BB 1297 enthalten und muss separat erworben werden. Zum Einbauen oder Austauschen der Batterie gehen Sie folgendermaßen vor:

1. Setzen Sie eine neue Batterie im BB 1297 mit der positiven Seite der Batterie nach oben und der negativen Seite neben der gedruckten Beschaltungsplatte ein.
2. Das BB 1297 kann jetzt in die CPU eingesetzt werden. Zur Installation des BB 1297 gehen Sie wie oben beschrieben vor.

So ersetzen Sie die Batterie des BBs 1297:

1. Bauen Sie das BB 1297 wie oben beschrieben aus der CPU aus.
2. Entnehmen Sie die alte Batterie vorsichtig mit Hilfe eines kleinen Schraubendrehers. Schieben Sie die Batterie unter der Klemme heraus.
3. Setzen Sie eine neue Batterie CR1025 mit der positiven Seite der Batterie nach oben und der negativen Seite neben der gedruckten Beschaltungsplatte ein.
4. Bauen Sie das Batterieboard BB 1297 wieder wie oben beschrieben ein.



### 3.3.4 Einbau und Ausbau eines SMs

Tabelle 3- 6 Einbau eines SMs

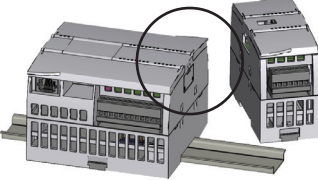
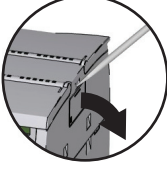
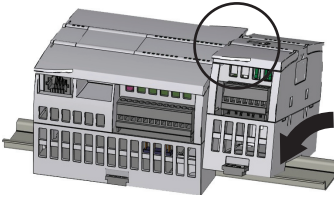
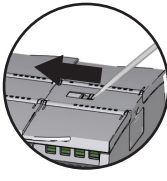
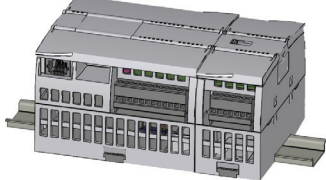
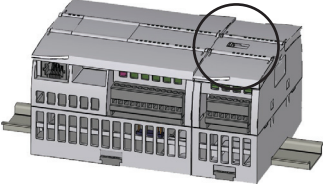
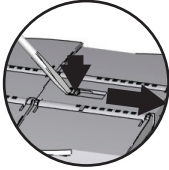
Aufgabenstellung	Vorgehensweise	
		<p>Bauen Sie Ihr SM nach der Montage der CPU ein.</p> <ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Nehmen Sie die Abdeckung für den Anschluss an der rechten Seite der CPU ab.</li> <li>3. Führen Sie einen Schraubendreher in den Schlitz über der Abdeckung ein.</li> <li>4. Hebeln Sie die Abdeckung vorsichtig oben aus ihrer Halterung und nehmen Sie sie ab. Bewahren Sie die Abdeckung zur späteren Verwendung auf.</li> </ol>
		<p>SM an die CPU anschließen:</p> <ol style="list-style-type: none"> <li>1. Positionieren Sie das SM neben der CPU.</li> <li>2. Hängen Sie das SM oben an der Hutschiene ein.</li> <li>3. Ziehen Sie die untere Hutschiene-Klemme heraus, damit das SM über die Schiene passt.</li> <li>4. Schwenken Sie das SM nach unten in die Position neben der CPU und drücken Sie die untere Klemme herunter, so dass das SM fest auf der Hutschiene sitzt.</li> </ol>
	<p>Der Busstecker stellt dadurch die mechanische und elektrische Verbindung für das SM her.</p> <ol style="list-style-type: none"> <li>1. Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des SM an.</li> <li>2. Schieben Sie die Lasche ganz nach links, um den Busstecker in die CPU zu schieben.</li> </ol> <p>Gehen Sie genauso vor, um ein Signalmodul an einem Signalmodul einzubauen.</p>	

Tabelle 3-7 Ausbau eines SMs

Aufgabenstellung	Vorgehensweise
	<p>Sie können jedes SM ausbauen, ohne die CPU oder andere SMs ausbauen zu müssen.</p> <ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Ziehen Sie die E/A-Steckverbinder und die Verdrahtung vom SM (Seite 60) ab.</li> <li>3. Ziehen Sie den Busstecker zurück.             <ul style="list-style-type: none"> <li>– Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des SM an.</li> <li>– Drücken Sie nach unten, um den Klemmenblock von der CPU zu lösen.</li> <li>– Schieben Sie die Lasche ganz nach rechts.</li> </ul> </li> </ol> <p>Ist auf der rechten Seite ein weiteres SM eingebaut, wiederholen Sie diese Vorgehensweise für das SM.</p>
	<p>Nehmen Sie das SM ab:</p> <ol style="list-style-type: none"> <li>1. Ziehen Sie die untere Hutschiene klemme heraus, damit das SM von der Schiene gelöst wird.</li> <li>2. Schwenken Sie das SM nach oben und von der Schiene herunter. Nehmen Sie das SM aus dem System heraus.</li> <li>3. Bringen Sie ggf. eine Abdeckung am Busstecker der CPU an, um Verschmutzung zu vermeiden.</li> </ol> <p>Gehen Sie genauso vor, um ein Signalmodul von einem Signalmodul auszubauen.</p>

### 3.3.5 Einbau und Ausbau eines CMs oder CPs

Schließen Sie die Kommunikationsmodule an die CPU an und bauen Sie alle Module gemeinsam ein. Dies wird unter Einbau und Ausbau der CPU (Seite 52) gezeigt.

Tabelle 3- 8 Einbau eines CMs oder CPs

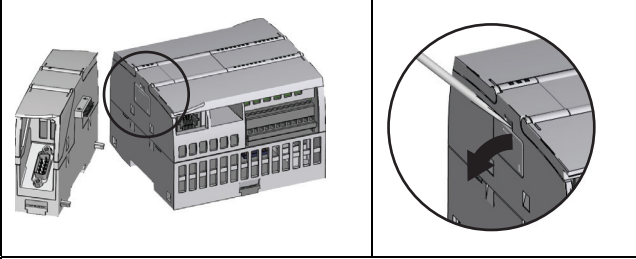
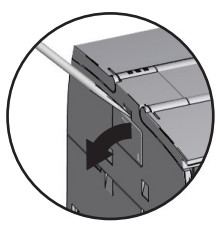

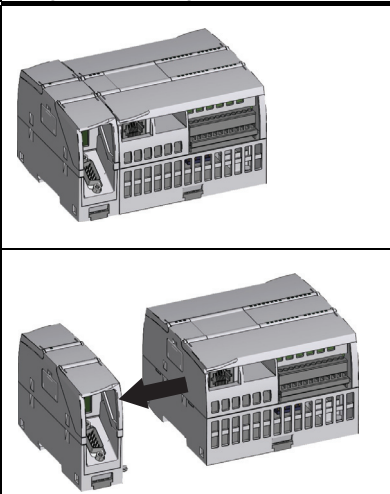
Aufgabenstellung		Vorgehensweise
		<ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Schließen Sie das CM an die CPU an und bauen Sie alle Module gemeinsam auf der Hutschiene oder in der Schalttafel ein.</li> <li>3. Nehmen Sie die Busabdeckung an der linken Seite der CPU ab: <ul style="list-style-type: none"> <li>– Führen Sie einen Schraubendreher in den Schlitz über der Busabdeckung ein.</li> <li>– Hebeln Sie die Abdeckung vorsichtig oben aus ihrer Halterung.</li> </ul> </li> <li>4. Nehmen Sie die Busabdeckung ab. Bewahren Sie die Abdeckung zur späteren Verwendung auf.</li> <li>5. CM oder CP an die CPU anschließen: <ul style="list-style-type: none"> <li>– Richten Sie den Busstecker und die Stifte des CMs zu den Bohrungen in der CPU aus.</li> <li>– Drücken Sie die Bauteile fest zusammen, bis die Stifte einrasten.</li> </ul> </li> <li>6. Bauen Sie die CPU und den CP auf einer DIN-Schiene oder in eine Schalttafel ein.</li> </ol>
		

Tabelle 3- 9 Ausbau eines CMs oder CPs

Aufgabenstellung	Vorgehensweise
	<p>Bauen Sie die CPU und das CM gemeinsam aus der Standard-Hutschiene bzw. der Schalttafel aus.</p> <ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Ziehen Sie die E/A-Steckverbinder, die Verdrahtung und die Kabel von der CPU und den CMs ab.</li> <li>3. Bringen Sie bei der Hutschienenmontage die unteren Hutschienenklemmen an der CPU und den CMs in die ausgezogene Stellung.</li> <li>4. Bauen Sie die CPU und die CMs aus der Standard-Hutschiene bzw. der Schalttafel aus.</li> <li>5. Halten Sie die CPU und die CMs fest und ziehen Sie sie auseinander.</li> </ol>

**VORSICHT**

Verwenden Sie für die Trennung der Module kein Werkzeug, da sonst die Teile beschädigt werden.

**3.3.6 Ausbau und Wiedereinbau des S7-1200 Klemmenblocks**

Die CPU, das SB und SM verfügen über abnehmbare Steckverbinder, um die Verdrahtung zu vereinfachen.

Tabelle 3- 10 Ausbau des Klemmenblocks

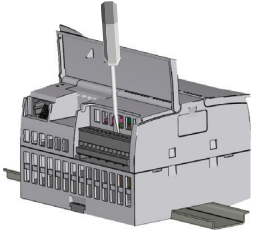
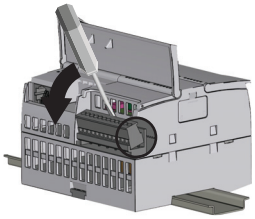
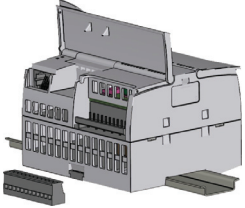
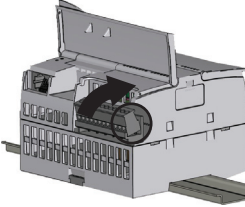
Aufgabenstellung	Vorgehensweise
	<p>Bereiten Sie das System für den Ausbau des Klemmenblocks vor, indem Sie die Spannung der CPU ausschalten und die Abdeckung oberhalb des Klemmenblocks öffnen.</p> <ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Suchen Sie auf der Oberseite des Klemmenblocks den Schlitz für die Spitze des Schraubendrehers.</li> <li>3. Führen Sie einen Schraubendreher in den Schlitz ein.</li> <li>4. Hebeln Sie die Oberseite des Klemmenblocks vorsichtig aus der Halterung in der CPU. Es ist ein Schnappgeräusch zu hören, wenn sich der Klemmenblock löst.</li> <li>5. Fassen Sie den Klemmenblock mit der Hand und ziehen Sie ihn aus der CPU heraus.</li> </ol>
	

Tabelle 3- 11 Einbau des Klemmenblocks

Aufgabenstellung	Vorgehensweise
	<p>Bereiten Sie die Komponenten für den Einbau des Klemmenblocks vor, indem Sie die Spannung der CPU ausschalten und die Abdeckung für den Steckverbinder öffnen.</p> <ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Richten Sie ihn zu den Stiften im Gerät aus.</li> <li>3. Richten Sie die Verdrahtungsseite des Klemmenblocks im Stecksockel richtig aus.</li> <li>4. Drücken Sie den Klemmenblock mit einer Drehbewegung fest nach unten, bis er einrastet.</li> </ol>
	<p>Prüfen Sie, ob der Klemmenblock richtig ausgerichtet ist und fest sitzt.</p>

### 3.3.7 Einbau und Ausbau des Erweiterungskabels

Das S7-1200 Erweiterungskabel sorgt für zusätzliche Flexibilität beim Konfigurieren des Layouts Ihres S7-1200 Systems. Pro CPU-System dürfen Sie ein Erweiterungskabel verwenden. Sie bauen das Erweiterungskabel entweder zwischen der CPU und dem ersten SM oder zwischen zwei SMs ein.

Tabelle 3- 12 Ein- und Ausbau des Steckers des Erweiterungskabels

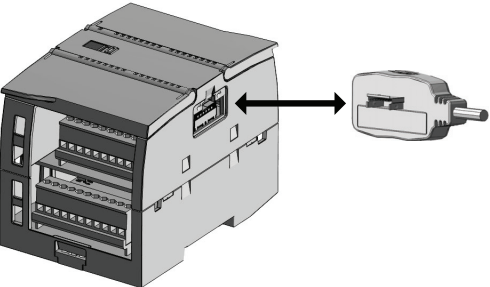
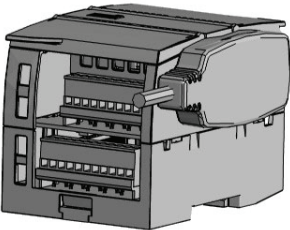
Aufgabenstellung	Vorgehensweise
	<p>Stecker einbauen:</p> <ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Drücken Sie den Steckverbinder in den Busanschluss auf der rechten Seite des Signalmoduls oder der CPU.</li> </ol> <p>Stecker ausbauen:</p> <ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Ziehen Sie den Stecker vom Signalmodul oder von der CPU ab.</li> </ol>
	

Tabelle 3- 13 Einbau der Buchse des Erweiterungskabels

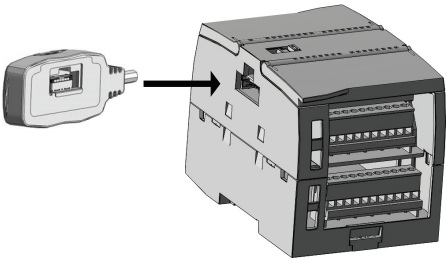
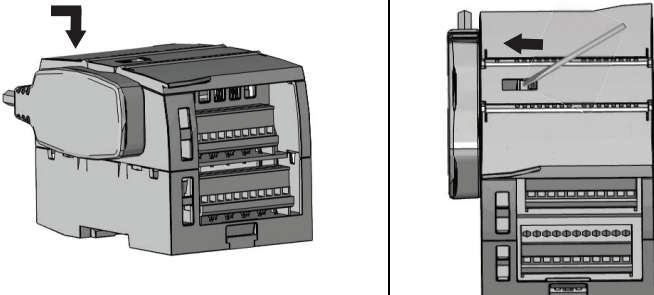
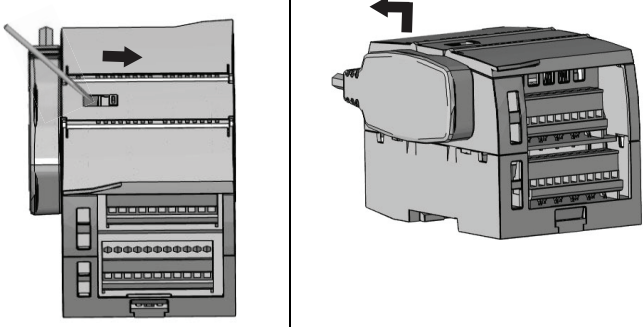
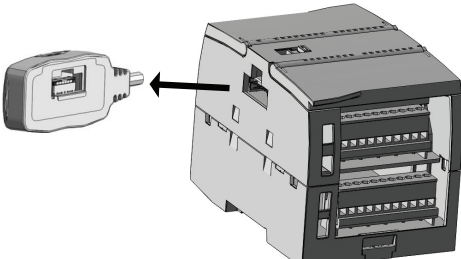
Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Stecken Sie die Buchse auf den Busanschluss an der linken Seite des Signalmoduls.</li> <li>3. Schieben Sie den Hakenfortsatz der Buchse in das Gehäuse am Busanschluss und drücken Sie vorsichtig, damit der Haken einrastet.</li> <li>4. Befestigen Sie die Steckverbindung:                             <ul style="list-style-type: none"> <li>- Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des Signalmoduls an.</li> <li>- Schieben Sie die Lasche ganz nach links.</li> </ul>                             Damit der Steckverbinder einrastet, müssen Sie die Lasche des Steckverbinders ganz nach links schieben. Die Steckverbinderlasche muss vollständig einrasten.                         </li> </ol>
	

Tabelle 3- 14 Ausbau der Buchse des Erweiterungskabels

Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> <li>1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.</li> <li>2. Lösen Sie den Steckverbinder:                             <ul style="list-style-type: none"> <li>- Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des Signalmoduls an.</li> <li>- Drücken Sie ihn leicht nach unten und schieben Sie die Lasche ganz nach rechts.</li> </ul> </li> <li>3. Heben Sie den Steckverbinder leicht an, damit sich der Haken löst.</li> <li>4. Nehmen Sie die Buchse ab.</li> </ol>
	

### 3.3.8 TS (Teleservice)-Adapter

#### 3.3.8.1 Anschließen des TeleService-Adapters

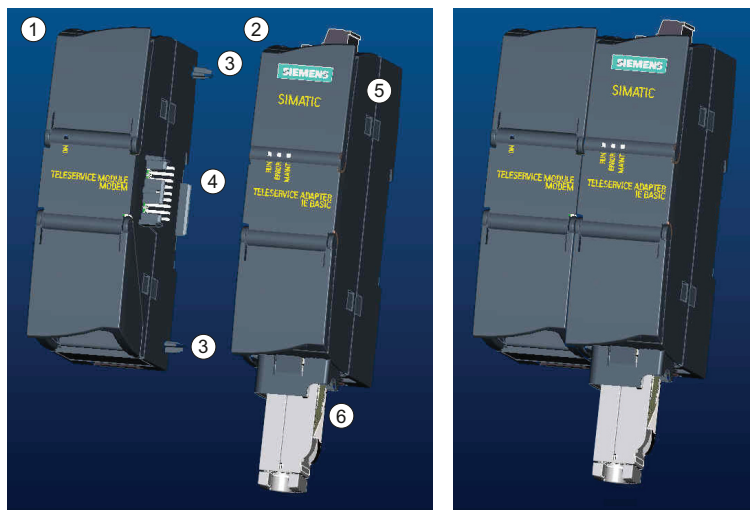
Bevor Sie den modularen TS (Teleservice)-Adapter IE Basic einbauen, müssen Sie zunächst den TS-Adapter und ein TS-Modul anschließen.

Verfügbare TS-Module:

- TS-Modul RS232
- TS-Modul Modem
- TS-Modul GSM
- TS-Modul ISDN

#### VORSICHT

Das TS-Modul kann beschädigt werden, wenn Sie die Kontakte des Steckverbinders ④ des TS-Moduls berühren. Befolgen Sie die EGB-Schutzmaßnahmen, um eine Beschädigung des TS-Moduls durch elektrostatische Entladung zu verhindern. Vergewissern Sie sich, bevor Sie ein TS-Modul und einen TS-Adapter anschließen, dass sich beide im Ruhezustand befinden.



- |   |            |   |                              |
|---|------------|---|------------------------------|
| ① | TS-Modul   | ④ | Steckverbinder des TS-Moduls |
| ② | TS-Adapter | ⑤ | Kann nicht geöffnet werden   |
| ③ | Elemente   | ⑥ | Ethernet-Port                |

**VORSICHT**

Stellen Sie vor dem Anschließen eines TS-Moduls und eines TS-Adapters sicher, dass die Kontaktstifte ④ nicht verbogen sind. Stellen Sie beim Anschließen sicher, dass der Stecker und die Führungselemente korrekt positioniert sind.

Schließen Sie nur ein TS-Modul an den TS-Adapter an. Stecken Sie den TS-Adapter nicht mit Gewalt auf ein anderes Gerät, z. B. eine S7-1200 CPU. Ändern Sie nicht die mechanische Konstruktion des Steckverbinders und entfernen oder beschädigen Sie nicht die Führungselemente.

**3.3.8.2 Einlegen der SIM-Karte**

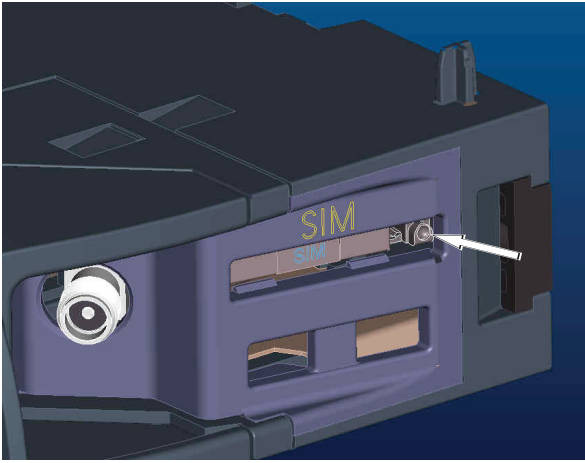
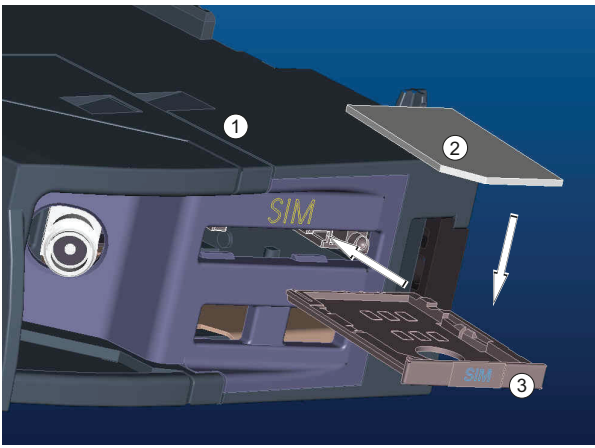
Lokalisieren Sie den SIM-Kartensteckplatz auf der Unterseite des TS module GSM.

**ACHTUNG**

Die SIM-Karte darf nur entnommen oder gesteckt werden, wenn das TS module GSM ausgeschaltet ist.



Tabelle 3- 15 Einlegen der SIM-Karte

Vorgehensweise	Aufgabenstellung							
<p>Drücken Sie mit einem spitzen Objekt auf den Auswurfknopf des SIM-Kartenfachs (in Pfeilrichtung) und nehmen Sie das SIM-Kartenfach heraus.</p>								
<p>Legen Sie die SIM-Karte wie gezeigt in das SIM-Kartenfach ein und stecken Sie das SIM-Kartenfach wieder in den Steckplatz.</p>		<table border="1"> <tr> <td data-bbox="1155 893 1219 927">①</td> <td data-bbox="1225 893 1476 927">TS Module GSM</td> </tr> <tr> <td data-bbox="1155 936 1219 969">②</td> <td data-bbox="1225 936 1476 969">SIM-Karte</td> </tr> <tr> <td data-bbox="1155 978 1219 1012">③</td> <td data-bbox="1225 978 1476 1012">SIM-Kartenfach</td> </tr> </table>	①	TS Module GSM	②	SIM-Karte	③	SIM-Kartenfach
①	TS Module GSM							
②	SIM-Karte							
③	SIM-Kartenfach							

**Hinweis**

Stellen Sie sicher, dass die SIM-Karte in der richtigen Richtung ins Kartenfach eingelegt wird. Andernfalls stellt die Karte keine Verbindung mit dem Modul her und das Kartenfach lässt sich möglicherweise nicht mehr über den Auswurfknopf herausnehmen.

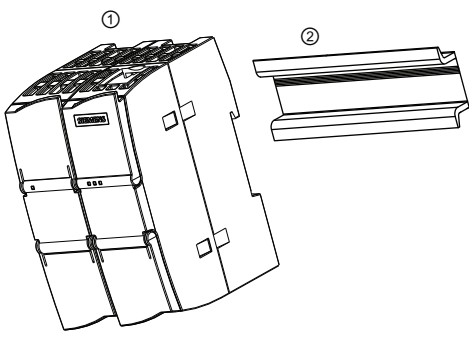
**3.3.8.3 Einbau des TS-Adapters**


Voraussetzungen: Sie müssen den TS-Adapter und das TS-Modul miteinander verbunden haben und die DIN-Schiene muss installiert sein.

**Hinweis**

Wenn Sie das TS-Gerät vertikal oder in Umgebungen mit starken Schwingungen einbauen, kann sich das TS-Modul vom TS-Adapter lösen. Montieren Sie eine Endhalterung 8WA1 808 auf der DIN-Schiene, um sicherzustellen, dass die Module miteinander verbunden bleiben.

Tabelle 3- 16 Einbau und Ausbau des TS-Adapters

Aufgabenstellung	Vorgehensweise
	<p>Einbau:</p> <ol style="list-style-type: none"> <li>1. Haken Sie den TS-Adapter mit dem angeschlossenen TS-Modul ① auf der DIN-Schiene ein ②.</li> <li>2. Drehen Sie das Gerät nach hinten, bis es einrastet.</li> <li>3. Drücken Sie den DIN-Schienen-Clip an jedem Modul hinein, um die Module auf der DIN-Schiene zu befestigen.</li> </ol> <p>Ausbau:</p> <ol style="list-style-type: none"> <li>1. Ziehen Sie das Analogkabel und das Ethernet-Kabel von der Unterseite des TS-Adapters ab.</li> <li>2. Schalten Sie die Spannungsversorgung des TS-Adapters aus.</li> <li>3. Lösen Sie die DIN-Schienen-Clips an beiden Modulen mit einem Schraubendreher.</li> <li>4. Drehen Sie das Gerät nach oben, um es von der DIN-Schiene abzunehmen.</li> </ol>

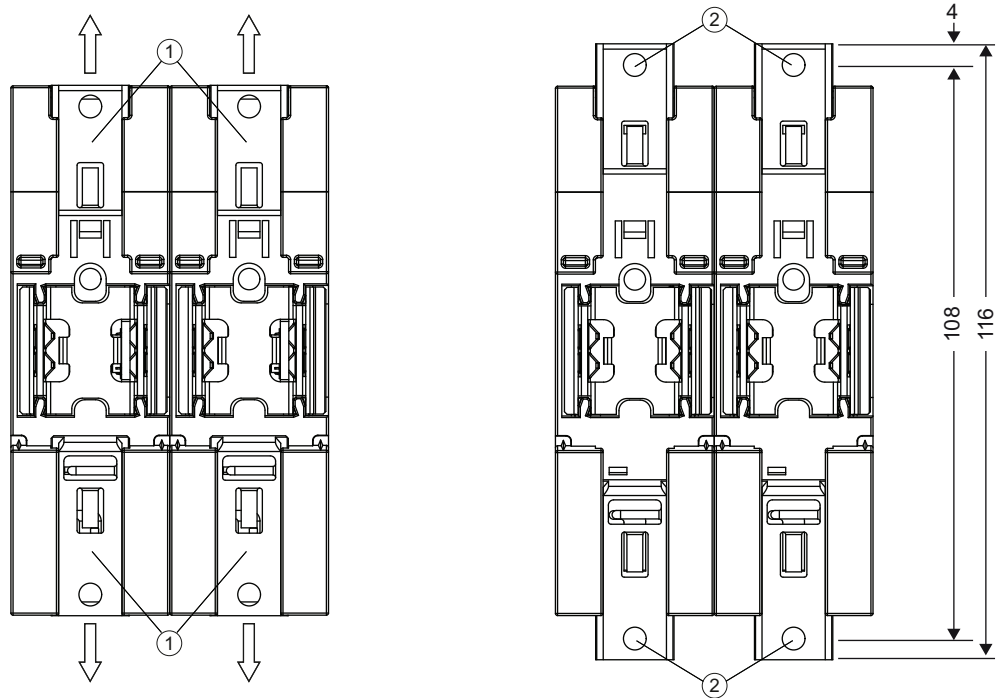
 <b>WARNUNG</b>
<p>Bevor Sie die Versorgung des Geräts ausschalten, heben Sie die Erdung des TS-Adapters auf, indem Sie das Analogkabel und das Ethernet-Kabel abziehen.</p>

**3.3.8.4 Wandmontage des TS-Adapters**

Voraussetzungen: Sie müssen den TS-Adapter und das TS-Modul miteinander verbunden haben.

1. Schieben Sie den Schieber für Anbauten ① zur Rückseite des TS-Adapters und des TS-Moduls in Pfeilrichtung, bis der Schieber einrastet.
2. Schrauben Sie den TS-Adapter und das TS-Modul an der mit ② gekennzeichneten Position an der gewünschten Montagewand fest.

Die folgende Abbildung zeigt den TS-Adapter von hinten, mit den Schiebern ① in beiden Positionen:



- ① Schieber für Anbauten
- ② Bohrungen zur Wandmontage

### 3.4 Verdrahtungsrichtlinien

Die ordnungsgemäße Erdung und Verdrahtung aller elektrischen Geräte ist wichtig für den optimalen Betrieb Ihres Systems und für zusätzliche Störfestigkeit für Ihre Anwendung und die S71200. Ausführliche Informationen finden Sie in den technischen Daten (Seite 743) der S7-1200 Schaltpläne.

#### Voraussetzung

Bevor Sie ein elektrisches Gerät erden oder verdrahten, müssen Sie sicherstellen, dass die Spannungsversorgung der Geräte ausgeschaltet ist. Achten Sie außerdem darauf, dass auch alle angeschlossenen Geräte ausgeschaltet sind.

Stellen Sie sicher, dass Sie bei der Verdrahtung der S7-1200 und aller angeschlossenen Geräte alle geltenden und verbindlichen Normen befolgen. Beachten Sie bei Einbau und Betrieb der Geräte die entsprechenden nationalen und regionalen Vorschriften. Erfragen Sie bei den Behörden vor Ort die Normen und Vorschriften, die für Ihren speziellen Fall zu befolgen sind.

 **WARNUNG**

Wenn Sie die S7-1200 oder daran angeschlossene Geräte in eingeschaltetem Zustand einbauen oder verdrahten, kann es passieren, dass Sie einen elektrischen Schlag bekommen oder die Geräte unerwartet arbeiten. Ist die Spannungsversorgung der S7-1200 und aller daran angeschlossenen Geräte während des Einbaus bzw. Ausbaus von Geräten nicht abgeschaltet, so kann dies aufgrund von elektrischem Schlag oder unerwartetem Betrieb der Geräte zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, dass vor dem Einbau bzw. Ausbau eines Geräts die Spannungsversorgung der S71200 abgeschaltet ist.

Denken Sie beim Planen von Erdung und Verdrahtung Ihres S7-1200 Systems immer an die Sicherheit. Elektronische Steuerungsgeräte wie die S7-1200 können ausfallen und dadurch unerwarteten Betrieb der gesteuerten oder beobachteten Geräte hervorrufen. Deshalb sollten Sie Sicherheitseinrichtungen implementieren, die von der S7-1200 unabhängig sind und vor möglichen Personen- und/oder Sachschäden schützen.

 **WARNUNG**

Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch den unkontrollierten Betrieb der gesteuerten Geräte verursachen. Unvorhersehbarer Betrieb kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.


Sorgen Sie daher für eine NOTAUSfunktion, elektromechanische oder andere redundante Sicherheitseinrichtungen, die von Ihrer S7-1200 unabhängig sind.

### Richtlinien für die Potentialtrennung

Die Grenzwerte der AC-Spannungsversorgung und die E/A-Grenzen zu AC-Stromkreisen der S7-1200 sind dafür konzipiert und zugelassen, sichere elektrische Trennung zwischen AC-Leitungsspannungen und Niederspannungskreisen zu bieten. Je nach Norm umfassen diese Grenzen doppelte oder verstärkte Isolierung bzw. grundlegende plus zusätzliche Isolierung. Komponenten, die diese Grenzen kreuzen, z.B. optische Koppler, Kondensatoren, Transformatoren und Relais, haben die Zulassung, sichere elektrische Trennung zu bieten. Potentialtrennungsgrenzen, die diese Anforderungen erfüllen, werden in den Produktdatenblättern der S7-1200 mit einer elektrischen Trennung von mindestens 1500 V AC angegeben. Diese Kennzeichnung beruht auf einer Routineprüfung im Werk von (2Ue + 1000 V AC) oder äquivalent gemäß zugelassenen Verfahren. Die Grenzwerte für die sichere elektrische Trennung der S7-1200 wurden typgeprüft nach 4242 V DC.

Geberversorgungsausgang, Kommunikationsstromkreise und Stromkreise der internen Logik einer S7-1200 mit integrierter AC-Spannungsversorgung werden als Sicherheitskleinspannung (SELV) nach EN 61131-2 gespeist.

Um die sichere Eigenschaft der Niederspannungskreise der S7-1200 zu erhalten, müssen externe Anschlüsse an Kommunikationsports, analoge Stromkreise sowie sämtliche 24-V-Nennspannungsversorgung und alle E/A-Stromkreise aus zugelassenen Quellen gespeist werden, die die Anforderungen nach verschiedenen Normen für SELV, PELV, Klasse 2, Spannungsbegrenzt oder Leistungsbegrenzt erfüllen.

 <b>WARNUNG</b>
<p>Der Einsatz nicht potentialgetrennter oder einfach isolierter Versorgungen für die Niederspannungskreise aus AC-Leitungen kann zu gefährlichen Spannungen in Kreisen führen, die als berührungssicher gelten, z.B. Kommunikationskreise oder Niederspannungsgeberverdrahtung.</p> <p>Unerwartet hohe Spannungen können einen elektrischen Schlag verursachen, der zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.</p> <p>Verwenden Sie nur solche Hochspannungs-/Niederspannungswandler, die als Quellen berührungssicherer Kreise mit begrenzter Spannung zugelassen sind.</p>

### Richtlinien für die Erdung der S7-1200

Am besten erden Sie Ihre Anwendung, indem Sie darauf achten, dass alle gemeinsamen Anschlüsse und alle Erdanschlüsse Ihrer S7-1200 und aller angeschlossenen Geräte an einer einzigen Stelle geerdet werden. Diese Stelle muss direkt mit der Systemerde verbunden werden.

Erdleitungen sollten möglichst kurz sein und einen großen Aderquerschnitt, z. B. 2 mm<sup>2</sup> (14 AWG) haben.

Beachten Sie beim Auswählen von Erdungspunkten die entsprechenden Sicherheitsvorschriften, und stellen Sie die einwandfreie Funktion von stromkreisunterbrechenden Schutzeinrichtungen sicher.

### Richtlinien für die Verdrahtung der S71200

Wenn Sie die Verdrahtung Ihrer S7-1200 planen, richten Sie einen Einzeltrennschalter ein, der gleichzeitig die Spannung der Spannungsversorgung für die S7-1200 CPU, die Spannung aller Eingangskreise und die Spannung aller Ausgangskreise trennt. Sorgen Sie für Überstromschutz, z. B. durch eine Sicherung oder einen Schutzschalter, um Fehlerstrom in der Versorgungsverdrahtung zu begrenzen. Sie können zusätzlichen Schutz durch Sicherungen oder andere Strombegrenzungen in den einzelnen Ausgangskreisen implementieren.

Versehen Sie blitzschlaggefährdete Leitungen mit einem geeigneten Überspannungsschutz.

Vermeiden Sie es, Niederspannungssignalleitungen und Kommunikationskabel in der gleichen Kabelbahn wie AC-Versorgungsleitungen und schnellschaltende Hochgeschwindigkeits-DC-Leitungen zu verlegen. Leitungen sollten Sie paarweise verlegen: den Neutral oder Nullleiter zusammen mit dem Phasenleiter oder der Signalleitung.

Verlegen Sie Leitungen so kurz wie möglich und achten Sie darauf, dass der Leitungsquerschnitt dem benötigten Strom entspricht. Am Klemmenblock von CPU und SM können Drahtquerschnitte von 2 mm<sup>2</sup> bis 0,3 mm<sup>2</sup> (14 AWG bis 22 AWG) angeschlossen werden. Am Klemmenblock des SBs können Drahtquerschnitte von 1,3 mm<sup>2</sup> bis 0,3 mm<sup>2</sup> (16 AWG bis 22 AWG) angeschlossen werden. Verwenden Sie als optimalen Schutz vor Störfestigkeit geschirmte Leitungen. Typischerweise erreichen Sie durch Erdung des Schirms an der S7-1200 die besten Ergebnisse.

Wenn Sie einen Eingangskreis verdrahten, der durch eine externe Spannungsversorgung versorgt wird, implementieren Sie ein Überstromschutzgerät in dem Eingangskreis. Bei Kreisen, die durch die 24-V-DC-Geberversorgung der S71200 versorgt werden, ist kein externer Schutz erforderlich, weil die Geberversorgung bereits strombegrenzt ist.

Alle S7-1200 Module verfügen über steckbare Klemmenblöcke für die Anwenderverdrahtung. Achten Sie darauf, dass der Klemmenblock fest sitzt und die Leitungen fest im Klemmenblock verdrahtet sind, damit es nicht zu losen Anschlüssen kommt. Ziehen Sie die Schrauben jedoch nicht zu fest an, damit Sie den Klemmenblock nicht beschädigen. Das maximale Drehmoment für die Schrauben des Klemmenblocks von CPU und SM beträgt 0,56 Nm. Das maximale Drehmoment für die Schrauben des Klemmenblocks des SBs beträgt 0,33 Nm.

Die S7-1200 arbeitet mit Potentialtrennungsgrenzen, wodurch das Entstehen unerwünschter Ströme in Ihrer Anlage verhindert wird. Wenn Sie die Verdrahtung für Ihr System planen, berücksichtigen Sie diese Potentialtrennungsgrenzen. Im Kapitel Technische Daten finden Sie die Werte für die zur Verfügung gestellte Potentialtrennung und die Anordnung der Potentialtrennungsgrenzen. Verlassen Sie sich nicht auf Potentialtrennungsgrenzen mit Nennwerten unter 1500 V AC als Sicherheitsgrenzen.

### Richtlinien für Lampenlasten

Lampenlasten schädigen Relaiskontakte aufgrund des hohen Einschaltstoßstroms. Dieser Stoßstrom ist nominal 10 bis 15 Mal so hoch wie der stationäre Strom einer Wolframlampe. Für Lampenlasten, die während der Lebensdauer der Anwendung sehr häufig geschaltet werden, wird ein austauschbares Koppelrelais oder ein Stoßstrombegrenzer empfohlen.

### Richtlinien für induktive Lasten

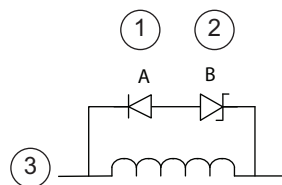
Versehen Sie induktive Lasten mit Schutzbeschaltungen, die den Spannungsanstieg beim Ausschalten des Steuerungsausgangs begrenzen. Schutzbeschaltungen schützen Ihre Ausgänge vor frühzeitigem Ausfall aufgrund hoher Spannungen beim Ausschalten induktiver Lasten. Außerdem begrenzen Schutzbeschaltungen die elektrischen Störungen, die beim Schalten induktiver Lasten entstehen. Am effektivsten verringern Sie elektrische Störungen durch Anordnen einer externen Schutzbeschaltung parallel zur Last und in der Nähe der Last.

Die DC-Ausgänge der S7-1200 umfassen interne Schutzbeschaltungen, die für die induktiven Lasten in den meisten Anwendungen adäquat sind. Da die Relais-Ausgangskontakte der S7-1200 zum Schalten einer DC-Last oder einer AC-Last verwendet werden können, ist kein interner Schutz vorhanden.

### Hinweis

Die Effektivität einer Schutzbeschaltung hängt von der jeweiligen Anwendung ab und muss immer für den Einzelfall geprüft werden. Es müssen immer alle Komponenten einer Schutzbeschaltung für den Einsatz in der Anwendung bemessen sein.

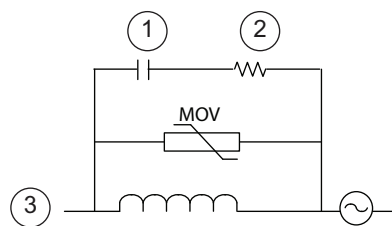
### Typische Schutzbeschaltungen für DC- oder Relaisausgänge, die induktive DC-Lasten schalten



- ① Diode 1N4001 oder gleichwertig
- ② Zener-Diode 8,2 V (DC-Ausgänge),  
Zener-Diode 36 V (Relaisausgänge)
- ③ Ausgang

In den meisten Anwendungen ist der zusätzliche Einsatz einer Diode (A) parallel zur induktiven DC-Last geeignet, doch wenn Ihre Anwendung schnellere Ausschaltzeiten benötigt, ist der Einsatz einer Zener-Diode (B) empfehlenswert. Bemessen Sie die Zener-Diode gemäß dem Strom im Ausgangskreis.

### Typische Schutzbeschaltungen für Relaisausgänge, die induktive AC-Lasten schalten



- ① 0,1  $\mu\text{F}$
- ② 100 bis 120  $\Omega$
- ③ Ausgang

Wenn Sie mit einem Relaisausgang Lasten von 115 V/230 V AC schalten, ordnen Sie einen entsprechend bemessenen Widerstands-/Kondensator-Metalloxidvaristor (MOV) parallel zur AC-Last an. Achten Sie darauf, dass die Arbeitsspannung des Varistors mindestens 20 % höher ist als die Nennspannung.





## 4.1 Ausführung des Anwenderprogramms

Die CPU unterstützt die folgenden Bausteinarten für den Aufbau einer geeigneten Struktur Ihres Anwenderprogramms:

- Organisationsbausteine (OBs) legen die Struktur des Programms fest. Für einige OBs gibt es vordefiniertes Verhalten und Startereignisse, Sie können aber auch OBs mit Ihren eigenen Startereignissen anlegen. Die Bereiche der gültigen OB-Nummern finden Sie unter Prioritäten und Warteschlange für die Ausführung von Ereignissen (Seite 81).
- Funktionen (FCs) und Funktionsbausteine (FBs) enthalten den Programmcode, der den jeweiligen Aufgaben oder Parametrierungen entspricht. Jede FC bzw. jeder FB stellt eine Anzahl Ein- und Ausgangsparameter für die gemeinsame Nutzung der Daten mit dem aufrufenden Baustein bereit. Ein FB verwendet ferner einen weiteren Datenbaustein (Instanz-DB) für die Speicherung von Zustandswerten während der Ausführung, die von anderen Bausteinen im Programm verwendet werden können. Gültige FC- und FB-Nummern liegen im Bereich von 1 bis 65.535.
- Datenbausteine (DBs) speichern Daten, die von den Programmbausteinen verwendet werden können. Gültige DB-Nummern liegen im Bereich von 1 bis 65.535.

Die Ausführung des Anwenderprogramms beginnt mit einem oder mehreren optionalen Anlauf-OBs, die nach dem Wechsel in den Betriebszustand RUN einmal abgearbeitet werden, gefolgt von einem oder mehreren Zyklus-OBs, die zyklisch abgearbeitet werden. Ein OB kann auch einem Alarmereignis zugeordnet werden, das ein Standard- oder ein Fehlerereignis sein kann; er wird dann ausgeführt, wenn das entsprechende Ereignis eintritt.

Eine Funktion (FC) oder ein Funktionsbaustein (FB) ist ein Baustein mit Programmcode, der aus einem OB oder einer anderen FC oder einem FB aufgerufen werden kann. Folgende Schachtelungstiefen sind dabei möglich:

- 16 aus Zyklus- oder Anlauf-OBs
- 4 aus Verzögerungsalarm-, Weckalarm-, Uhrzeitalarm-, Prozessalarm-, Zeitfehler- oder Diagnosefehler-OBs


FCs sind keinem bestimmten Datenbaustein (DB) zugeordnet, während FBs direkt mit einem DB verbunden sind und diesen für die Übertragung von Parametern und die Speicherung von Zwischenwerten und -ergebnissen nutzen.

Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU begrenzt. Die Anzahl der einzelnen OBs, FCs, FBs und DBs ist nicht begrenzt. Die Gesamtzahl der Bausteine darf jedoch 1024 nicht überschreiten.

Jeder Zyklus umfasst das Schreiben der Ausgänge, das Lesen der Eingänge, das Bearbeiten der Anweisungen des Anwenderprogramms und die Durchführung der Hintergrundverarbeitung. Der Zyklus wird auch als Abtastzyklus oder Abtastung bezeichnet.

Die Module (SM, SB, BB, CB, CM oder CP) werden erkannt und erst beim Anlauf angemeldet.

- Das Stecken oder Ziehen eines Moduls im zentralen Baugruppenträger bei eingeschaltetem Gerät ist nicht möglich. Stecken oder ziehen Sie niemals ein Modul im zentralen Baugruppenträger, wenn die CPU eingeschaltet ist.

 <b>WARNUNG</b>
Das Stecken oder Ziehen eines Moduls (SM, SB, BB, CD, CM oder CP) im zentralen Baugruppenträger bei eingeschalteter CPU kann unvorhersehbares Verhalten verursachen, was wiederum zu Sachschaden und/oder Verletzungen führen kann. Sorgen Sie stets dafür, dass die CPU und der zentrale Baugruppenträger spannungslos sind, bevor Sie ein Modul im zentralen Baugruppenträger stecken oder ziehen.

- Eine SIMATIC Memory Card können Sie bei eingeschalteter CPU stecken oder ziehen. Das Stecken oder Ziehen einer Memory Card im Betriebszustand RUN der CPU verursacht jedoch, dass die CPU in STOP geht.

<b>VORSICHT</b>
Durch Stecken oder Ziehen einer Memory Card im Betriebszustand RUN der CPU geht die CPU in den Betriebszustand STOP, was zu Sachschaden an den gesteuerten Geräten oder im gesteuerten Prozess führen kann. Wenn Sie eine Memory Card stecken oder ziehen, geht die CPU sofort in den Betriebszustand STOP. Stellen Sie vor dem Stecken oder Ziehen einer Memory Card stets sicher, dass die CPU nicht aktiv eine Maschine oder einen Prozess steuert. Installieren Sie immer einen NOT-AUS-Schaltkreis für Ihre Anwendung bzw. Ihren Prozess.

- Wenn Sie in einem dezentralen E/A-Baugruppenträger (PROFINET oder PROFIBUS) im Betriebszustand RUN der CPU ein Modul stecken oder ziehen, generiert die CPU einen Eintrag im Diagnosepuffer und bleibt im Betriebszustand RUN.

Bei der Standardkonfiguration werden alle lokalen digitalen und analogen Ein- und Ausgänge synchron zum Zyklus mit einem internen Speicherbereich, dem so genannten Prozessabbild, aktualisiert. Das Prozessabbild enthält ein Momentabbild der physischen Ein- und Ausgänge (physische E/A von CPU, Signalboard und Signalmodulen).

Die CPU führt die folgenden Aufgaben durch:

- Die CPU schreibt die Ausgänge aus dem Prozessabbild der Ausgänge in die physischen Ausgänge.
- Die CPU liest die physischen Eingänge unmittelbar vor der Ausführung des Anwenderprogramms und speichert die Eingangswerte im Prozessabbild der Eingänge. Dadurch wird sichergestellt, dass diese Werte während der Ausführung der Anwenderanweisungen konsistent bleiben.
- Die CPU führt die Logik der Anwenderanweisungen durch und aktualisiert die Ausgangswerte im Prozessabbild der Ausgänge, statt in die tatsächlichen physischen Ausgänge zu schreiben.

Dieser Vorgang sorgt während der gesamten Ausführung der Anwenderanweisungen in dem jeweiligen Zyklus für eine konsistente Logik und verhindert ein Pendeln der physischen Ausgänge mit mehrmaligen Zustandswechseln im Prozessabbild der Ausgänge.

Sie können angeben, ob digitale und analoge E/A im Prozessabbild automatisch aktualisiert und gespeichert werden sollen. Wenn Sie ein Modul in die Gerätesicht einfügen, befinden sich dessen Daten im Prozessabbild der CPU (Standard). Die CPU führt den Datenaustausch zwischen dem Modul und dem Prozessabbild automatisch während der Aktualisierung des Prozessabbilds durch. Um digitale oder analoge Eingänge oder Ausgänge aus der automatischen Aktualisierung des Prozessabbilds zu entfernen, wählen Sie das entsprechende Gerät in der Gerätekonfiguration, rufen das Register "Eigenschaften" auf, erweitern es ggf., damit die gewünschten Eingänge und Ausgänge angezeigt werden, und wählen dann "E/A-Adressen/HW-Kennung". Ändern Sie dann den Eintrag für "Prozessabbild:" von "Zyklisches PA" in "---". Um die Eingänge und Ausgänge wieder in die automatische Aktualisierung des Prozessabbilds aufzunehmen, stellen Sie für diese Option wieder "Zyklisches PA" ein.

Sie können bei Ausführung einer Anweisung die Werte physischer Eingänge direkt lesen und auch direkt Werte in physische Ausgänge schreiben. Beim direkten Lesen wird auf den aktuellen Zustand des physischen Eingangs zugegriffen. Das Prozessabbild der Eingänge wird, unabhängig davon, ob der Eingang für die Speicherung im Prozessabbild konfiguriert ist, nicht aktualisiert. Beim direkten Schreiben in den physischen Ausgang werden sowohl das Prozessabbild der Ausgänge (sofern der Ausgang für die Speicherung im Prozessabbild konfiguriert ist) als auch der physische Ausgang aktualisiert. Hängen Sie die Endung ":P" an die E/A-Adresse an, wenn Sie möchten, dass das Programm direkt über den physischen Eingang bzw. Ausgang auf die E/A-Daten zugreift und nicht über das Prozessabbild.

Die CPU unterstützt dezentrale E/A in PROFINET- und PROFIBUS-Netzen (Seite 449).

### 4.1.1 Betriebszustände der CPU

Die CPU hat drei Betriebszustände: Betriebszustand STOP, Betriebszustand STARTUP und Betriebszustand RUN. Die Status-LEDs auf der Vorderseite der CPU geben den aktuellen Betriebszustand an.

- Im Betriebszustand STOP führt die CPU das Programm nicht aus. Sie können ein Projekt in die CPU laden.
- Im Betriebszustand STARTUP werden die Anlauf-OBs (sofern vorhanden) einmal ausgeführt. Alarmereignisse werden im Betriebszustand STARTUP nicht bearbeitet.
- Im Betriebszustand RUN werden die Programmzyklus-OBs wiederholt ausgeführt. Alarmereignisse können auftreten und an beliebigen Punkten im Betriebszustand RUN abgearbeitet werden. Einige Teile eines Projekts können im Betriebszustand RUN in die CPU geladen werden (Seite 735).

Die CPU unterstützt den Warmstart, um in den Betriebszustand RUN zu gehen. Während eines Warmstarts wird kein Umlöschen ausgeführt. Während eines Warmstarts werden alle nicht remanenten System- und Anwenderdaten initialisiert. Die remanenten Daten bleiben erhalten.

Beim Umräumen werden der Arbeitsspeicher sowie alle remanenten und nicht remanenten Speicherbereiche gelöscht und der Ladespeicher in den Arbeitsspeicher kopiert. Der Diagnosepuffer und die dauerhaft gespeicherten Werte der IP-Adresse werden beim Umräumen nicht gelöscht.

---

**Hinweis**

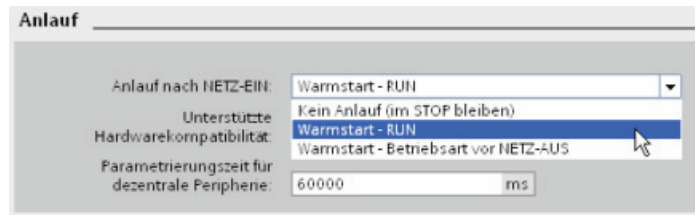
Wenn Sie einen oder mehrere DBs aus STEP 7 V11 in eine S7-1200 V2 CPU laden, werden die remanenten und nicht remanenten Werte dieser DBs auf ihre Startwerte gesetzt. Beim nächsten Wechsel nach RUN wird ein Warmstart durchgeführt, bei dem alle nicht remanenten Daten auf ihre Startwerte und alle remanenten Daten auf ihre gespeicherten Werte gesetzt werden.

Wenn Sie Projektelemente (wie die Gerätekonfiguration, Codebausteine oder DBs) aus STEP 7 V10.5 in eine beliebige S7-1200-CPU oder aus STEP 7 V11 in eine S7-1200 V1-CPU (oder eine V2-CPU, die als V1-CPU konfiguriert ist) laden, werden beim nächsten Wechsel in RUN **alle** DBs im Projekt auf die Startwerte zurückgesetzt.

---

Sie können die Einstellung "Anlauf nach NETZ-EIN" der CPU konfigurieren. Diese Einstellungen finden Sie in der Gerätekonfiguration der CPU unter "Anlauf". Beim Einschalten führt die CPU eine Reihe von Diagnoseprüfungen und anschließend die Systeminitialisierung durch. Während der Initialisierung des Systems löscht die CPU den gesamten nicht remanenten Speicherbereich der Merker und setzt alle nicht remanenten DB-Inhalte auf die Anfangswerte aus dem Ladespeicher zurück. Die CPU puffert den remanenten Speicherbereich der Merker und remanente DB-Inhalte und nimmt dann den entsprechenden Betriebszustand ein. Bestimmte Fehler verhindern, dass die CPU in den Betriebszustand RUN geht. Die CPU unterstützt die folgenden Konfigurationsoptionen:

- Kein Neustart (in STOP bleiben)
- Warmstart - RUN
- Warmstart - Betriebsart vor NETZ-AUS



### VORSICHT

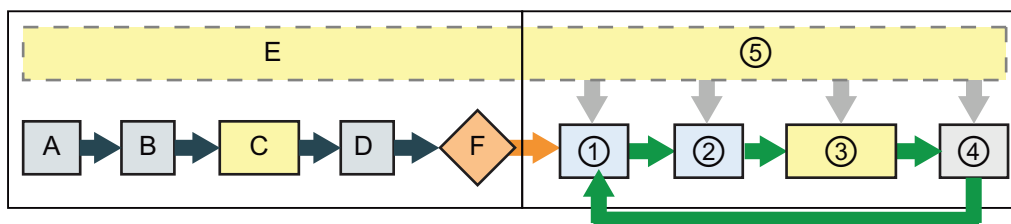
Die CPU kann aufgrund von behebbaren Fehlern in den Betriebszustand STOP gehen. Dies können z.B. der Ausfall eines ersetzbaren Signalmoduls oder temporäre Fehler sein wie Störungen der Netzleitung oder ein unvorhersehbares Anlaufereignis.

Wenn für die CPU die Einstellung "Warmstart - Betriebsart vor NETZ-AUS" konfiguriert ist, kehrt sie nach Behebung des Fehlers erst dann in den Betriebszustand RUN zurück, wenn sie einen neuen Befehl von STEP 7 erhält, um in den Betriebszustand RUN zu gehen. Ohne einen neuen Befehl behält die CPU den Betriebszustand STOP als den Zustand vor NETZ-AUS bei.

Für CPUs, die für den von einer STEP 7-Verbindung unabhängigen Betrieb vorgesehen sind, sollte üblicherweise die Einstellung "Warmstart - RUN" konfiguriert sein, damit die CPU nach Behebung eines Fehlerzustands durch Aus- und Wiedereinschalten wieder in den Betriebszustand RUN versetzt werden kann.

Sie können den aktuellen Betriebszustand mit den Befehlen "STOP" und "RUN" in den Online-Tools der Programmiersoftware ändern. Sie können ferner eine STP-Anweisung in Ihr Programm einfügen, um die CPU in den Betriebszustand STOP zu versetzen. Auf diese Weise können Sie die Bearbeitung Ihres Programms abhängig von der Programmlogik unterbrechen.

- Im Betriebszustand STOP bearbeitet die CPU Kommunikationsanforderungen und führt Selbstdiagnosen durch. Die CPU führt das Anwenderprogramm nicht aus und es finden keine automatischen Aktualisierungen des Prozessabbilds statt.  
Sie können Ihr Projekt nur laden, wenn sich die CPU im Betriebszustand STOP befindet.
- In den Betriebszuständen STARTUP und RUN führt die CPU die im folgenden Bild gezeigten Aufgaben aus.



**STARTUP**

- A Speicherbereich E (Abbild) wird gelöscht
- B Die Ausgänge werden mit dem letzten Wert oder dem Ersatzwert initialisiert
- C Die Anlauf-OBs werden ausgeführt
- D Der Zustand der physischen Eingänge wird in den Speicherbereich E kopiert
- E Alle Alarmereignisse werden in der Warteschlange für die Verarbeitung im Betriebszustand RUN gespeichert
- F Das Schreiben von Speicherbereich A in die physischen Ausgänge wird aktiviert

**RUN**

- ① Speicherbereich A wird in die physischen Ausgänge geschrieben
- ② Der Zustand der physischen Eingänge wird in den Speicherbereich E kopiert
- ③ Die Programmzyklus-OBs werden ausgeführt
- ④ Führt Selbstdiagnose durch
- ⑤ Alarme und Kommunikation werden in allen Teilen des Zyklus bearbeitet

**STARTUP-Verarbeitung**

Immer wenn der Betriebszustand von STOP nach RUN wechselt, löscht die CPU das Prozessabbild der Eingänge, initialisiert das Prozessabbild der Ausgänge und verarbeitet die Anlauf-OBs. Alle Lesezugriffe auf das Prozessabbild der Eingänge von Anweisungen in den Anlauf-OBs ergeben den Wert null und nicht den aktuellen Wert des physischen Eingangs. Deshalb müssen Sie, um den aktuellen Zustand eines physischen Eingangs im Betriebszustand STARTUP zu lesen, den Eingang direkt lesen. Dann werden die Anlauf-OBs und zugehörige FCs und FBs ausgeführt. Sind mehrere Anlauf-OBs vorhanden, so werden diese fortlaufend nach OB-Nummer, beginnend mit der niedrigsten OB-Nummer, abgearbeitet.

Jeder Anlauf-OB enthält Anlaufinformationen, damit Sie die Gültigkeit der remanenten Daten und der Echtzeituhr ermitteln können. Sie können in den Anlauf-OBs Anweisungen programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen. Die folgenden Anlaufadressen werden von Anlauf-OBs unterstützt:

Tabelle 4- 1 Vom Anlauf-OB unterstützte Anlaufadressen

Eingang	Datentyp	Beschreibung
LostRetentive	Bool	Dieses Bit ist wahr, wenn die Speicherbereiche der remanenten Daten verloren gegangen sind.
LostRTC	Bool	Dieses Bit ist wahr, wenn die Echtzeituhr verloren gegangen ist.

Die CPU führt während der Anlaufverarbeitung auch die folgenden Aufgaben aus.

- Alarme werden während der Anlaufphase in eine Warteschlange gestellt und nicht bearbeitet
- In der Anlaufphase findet keine Zykluszeitüberwachung statt
- Beim Anlaufen kann die Konfiguration der schnellen Zähler (HSC), der Impulsdauermodulation (PWM) und der Punkt-zu-Punkt-Kommunikationsmodule geändert werden
- Die eigentlichen Funktionsabläufe von HSC, PWM und PTP-Modulen finden nur im Betriebszustand RUN statt

Nach der Ausführung der Anlauf-OBs geht die CPU in den Betriebszustand RUN und bearbeitet die Steuerungsaufgaben in einem fortlaufenden Zyklus.

### Siehe auch

Zyklus anhalten (Seite 248)

Bedienpanel für die Online-CPU (Seite 723)

## 4.1.2 Bearbeitung des Zyklus im Betriebszustand RUN

In jedem Zyklus schreibt die CPU in die Ausgänge, sie liest die Eingänge, führt das Anwenderprogramm aus, aktualisiert die Kommunikationsmodule und antwortet auf Anwenderalarmereignisse und Kommunikationsanfragen. Kommunikationsanfragen werden während des Zyklus regelmäßig bearbeitet.

Diese Aktionen (außer den Anwenderalarmereignissen) werden zyklisch fortlaufend bearbeitet. Anwenderalarmereignisse, die aktiviert sind, werden nach der Priorität in der Reihenfolge ihres Auftretens bearbeitet.

Das System gewährleistet, dass der Zyklus innerhalb der maximalen Zykluszeit abgearbeitet wird, sonst wird ein Zeitfehler erzeugt.

- Jeder Zyklus beginnt mit der Abfrage der aktuellen Werte der digitalen und analogen Ausgänge im Prozessabbild und dem Schreiben dieser Werte in die physischen Ausgänge von CPU, SB und SMs, die für die automatische E/A-Aktualisierung konfiguriert sind (Standardkonfiguration). Greift eine Anweisung auf einen physischen Ausgang zu, so werden der Ausgang im Prozessabbild und der physische Ausgang aktualisiert.
- Im weiteren Verlauf des Zyklus werden die aktuellen Werte der digitalen und analogen Eingänge aus der CPU, der SB und den SMs, die für die automatische E/A-Aktualisierung konfiguriert sind (Standardkonfiguration), ausgelesen und diese Werte in das Prozessabbild geschrieben. Greift eine Anweisung auf einen physischen Eingang zu, so wird der Wert des physischen Eingangs geändert, der Eingang im Prozessabbild jedoch nicht aktualisiert.
- Nach dem Lesen der Eingänge wird das Anwenderprogramm von der ersten Anweisung bis zur letzten Anweisung ausgeführt. Darin enthalten sind alle Programmzyklus-OBs sowie alle zugehörigen FCs und FBs. Die Programmzyklus-OBs werden fortlaufend nach der OB-Nummer, beginnend mit der niedrigsten OB-Nummer, abgearbeitet.

Die Kommunikationsbearbeitung tritt während des Zyklus regelmäßig auf und unterbricht möglicherweise die Ausführung des Anwenderprogramms.

Zu den Selbstdiagnosen gehören regelmäßige Prüfungen des System und die Abfrage des Zustands der E/A-Module.

Alarmer können in jedem Teil des Zyklus auftreten, sie sind ereignisgesteuert. Tritt ein Ereignis auf, so unterbricht die CPU den Zyklus und ruft den OB für die Verarbeitung des Ereignisses auf. Wenn der OB das Ereignis abgearbeitet hat, setzt die CPU die Ausführung des Anwenderprogramms an der Stelle fort, an der es zuvor unterbrochen wurde.

### 4.1.3 Organisationsbausteine (OBs)

OBs steuern die Ausführung des Anwenderprogramms. Jedem OB muss eine eindeutige Nummer zugeordnet sein. Die Nummern unter 200 sind für bestimmte OBs belegt. Alle anderen OBs müssen eine Nummer über 200 besitzen.

Die Ausführung eines Organisationsbausteins wird durch bestimmte Ereignisse in der CPU angestoßen. OBs können sich nicht gegenseitig aufrufen und sie können auch nicht aus einer FC oder einem FB aufgerufen werden. Nur ein Startereignis, z. B. ein Diagnosealarm oder ein Zeitintervall, kann die Ausführung eines OBs auslösen. Die CPU bearbeitet die OBs dann entsprechend ihren Prioritätsklassen, wobei OBs mit höheren Prioritäten zuerst ausgeführt werden. Die niedrigste Prioritätsklasse ist 1 (für den Hauptprogrammzyklus), die höchste ist 26 (für Zeitfehler).

OBs steuern die folgenden Abläufe:

- Programmzyklus-OBs werden zyklisch ausgeführt, wenn die CPU im Betriebszustand RUN ist. Der Hauptbaustein des Programms ist ein Programmzyklus-OB. Er enthält die Anweisungen für die Steuerung Ihrer Anwendung und aus ihm heraus werden weitere Anwenderbausteine aufgerufen. Mehrere Programmzyklus-OBs sind zulässig, sie werden in numerischer Reihenfolge ausgeführt. OB 1 ist der Standardbaustein. Andere Programmzyklus-OBs müssen als OB 200 oder höher gekennzeichnet werden.
- Anlauf-OBs werden einmal ausgeführt, wenn der Betriebszustand der CPU von STOP nach RUN wechselt, beim Hochfahren in den Betriebszustand RUN und bei einem vorgegebenen Wechsel von STOP nach RUN. Anschließend beginnt die Ausführung des Zyklus-OBs. Es sind mehrere Anlauf-OBs zulässig. OB 100 ist der Standardbaustein. Alle anderen OBs müssen Nummern ab 200 haben.
- Weckalarm-OBs werden in bestimmten Abständen ausgeführt. Ein Weckalarm-OB unterbricht den Programmablauf in bestimmten, benutzerdefinierten Abständen, z. B. alle 2 Sekunden. Es können insgesamt bis zu 4 Verzögerungsereignisse und zyklische Ereignisse zu beliebigen Zeitpunkten konfiguriert werden, wobei für jedes konfigurierte Verzögerungsereignis oder zyklische Ereignis ein OB zulässig ist. Der OB muss die Nummer 200 oder höher haben.
- Prozessalarm-OBs werden ausgeführt, sobald das entsprechende Prozessereignis auftritt, z.B. steigende oder fallende Flanke an einem integrierten digitalen Eingang oder ein HSC-Ereignis. Ein Prozessalarm-OB unterbricht den normalen Programmablauf durch ein Signal eines Prozessereignisses. Sie definieren die Ereignisse in den Eigenschaften der Hardwarekonfiguration. Für jeden Prozessalarm ist ein OB zulässig. Der OB muss die Nummer 200 oder höher haben.



- Ein Zeitfehler-OB wird ausgeführt, wenn entweder die maximale Zykluszeit überschritten wird oder ein Zeitfehlerereignis auftritt. Die Verarbeitung von Zeitfehleralarmen wird von OB 80 durchgeführt. Wird dieser OB ausgelöst, beginnt die Ausführung, die den normalen Programmablauf oder auch einen anderen Ereignis-OB unterbricht. Die Ereignisse, die den Zeitfehleralarm und die Reaktion der CPU auf diese Ereignisse auslösen, werden im Folgenden beschrieben:
  - Überschreiten der maximalen Zykluszeit: Sie können die maximale Zykluszeit in den Eigenschaften der CPU konfigurieren. Wenn OB 80 nicht vorhanden ist, reagiert die CPU auf das Überschreiten der maximalen Zykluszeit, indem sie in STOP geht.
  - Zeitfehler: Wenn OB 80 nicht vorhanden ist, reagiert die CPU, indem sie in RUN bleibt. Zeitfehler treten auf, wenn ein Uhrzeitereignis verpasst oder wiederholt wird, wenn die Warteschlange überläuft oder ein Ereignis-OB (Zeitverzögerungsereignis, Uhrzeitereignis oder Weckalarm) startet, bevor die CPU die Ausführung des vorherigen OBs beendet hat.

Das Auftreten eines dieser Ereignisse erzeugt einen Eintrag im Diagnosepuffer, der das Ereignis beschreibt. Der Eintrag im Diagnosepuffer wird unabhängig davon erzeugt, ob OB 80 vorhanden ist oder nicht.

- Diagnosefehler-OBs werden ausgeführt, wenn ein Diagnosefehler erkannt und gemeldet wird. Ein Diagnose-OB unterbricht den normalen Programmablauf, wenn eine diagnosefähige Baugruppe einen Fehler erkennt (sofern für die Baugruppe der Diagnosealarm aktiviert wurde). Für Diagnosealarme ist ausschließlich OB 82 zulässig. Sie können eine Anweisung STP (CPU in STOP versetzen) in Ihren OB 82 einfügen, wenn Sie möchten, dass Ihre CPU bei einem Fehler dieser Art in den Betriebszustand STOP wechselt. Umfasst das Programm keinen Diagnose-OB, ignoriert die CPU den Fehler (und bleibt in RUN).

#### 4.1.4 Prioritäten und Warteschlange für die Ausführung von Ereignissen

Die CPU-Bearbeitung wird durch Ereignisse gesteuert. Ein Ereignis löst die Ausführung eines Alarm-OBs aus. Sie können während der Erstellung des Bausteins, während der Gerätekonfiguration oder über eine Anweisung ATTACH oder DETACH den Alarm-OB für ein Ereignis angeben. Einige Ereignisse wie das Programmzyklusereignis oder zyklische Ereignisse treten regelmäßig auf. Andere Ereignisse wie das Anlaufereignis oder Zeitverzögerungsereignisse treten einmalig auf. Einige Ereignisse treten auf, wenn es zu einer von der Hardware ausgelösten Veränderung kommt, z.B. ein Flankenereignis an einem Eingang oder ein Ereignis eines schnellen Zählers. Außerdem gibt es Ereignisse wie das Diagnosefehler- und das Zeitfehlerereignis, die nur im Fehlerfall auftreten. Die Ereignisprioritäten und Warteschlangen dienen zum Festlegen der Verarbeitungsreihenfolge der Alarm-OBs.

Das Programmzyklusereignis tritt einmal in jedem Programmzyklus auf. Während des Programmzyklus schreibt die CPU in die Ausgänge, liest die Eingänge und führt Programmzyklus-OBs aus. Das Programmzyklusereignis ist erforderlich und immer aktiviert. Für das Programmzyklusereignis haben Sie möglicherweise keinen Programmzyklus-OB oder Sie haben möglicherweise mehrere OBs. Nachdem das Programmzyklusereignis ausgelöst wurde, wird der Programmzyklus-OB mit der kleinsten Nummer ausgeführt (in der Regel OB 1). Die anderen Programmzyklus-OBs werden sequentiell (in numerischer Reihenfolge) innerhalb des Programmzyklus ausgeführt.

Die Weckalarmereignisse ermöglichen Ihnen, die Ausführung eines Alarm-OBs zu einer konfigurierten Zykluszeit einzurichten. Die anfängliche Zykluszeit wird konfiguriert, wenn der OB angelegt und als Weckalarm-OB eingerichtet wird. Ein zyklisches Ereignis unterbricht den Programmzyklus und führt den Weckalarm-OB aus (das zyklische Ereignis befindet sich in einer Klasse mit höherer Priorität als das Programmzyklusereignis).

Einem zyklischen Ereignis darf nur ein Weckalarm-OB zugeordnet werden.

Jedem zyklischen Ereignis kann eine Phasenverschiebung zugewiesen werden, so dass die Ausführung von Weckalarmen mit derselben Zykluszeit um den Wert der Phasenverschiebung verschoben werden kann. Der voreingestellte Wert der Phasenverschiebung ist 0. Um die anfängliche Phasenverschiebung zu ändern oder um die anfängliche Zykluszeit eines zyklischen Ereignisses zu ändern, klicken Sie in der Projektnavigation mit der rechten Maustaste auf den Weckalarm-OB, dann auf "Eigenschaften" und auf "Weckalarm". Geben Sie hier die neuen anfänglichen Werte ein. Sie können die Zykluszeit und Phasenverschiebung auch aus Ihrem Programm abfragen und ändern. Verwenden Sie dazu die Anweisungen Weckalarm-Parameter abfragen (QRY\_CINT) und Weckalarm-Parameter setzen (SET\_CINT). Die über die Anweisung SET\_CINT eingegebenen Werte für Zykluszeit und Phasenverschiebung werden bei Ausschalten oder Wechsel in STOP nicht gespeichert. Bei Wiedereinschalten oder Rückkehr in RUN werden wieder die ursprünglichen anfänglichen Werte verwendet. Die CPU unterstützt insgesamt vier Weckalarm- und Verzögerungsereignisse.

Das Anlaufereignis tritt einmal bei einem Wechsel von STOP in RUN auf und verursacht die Ausführung des Anlauf-OBs. Für das Anlaufereignis können mehrere OBs ausgewählt werden. Die Anlauf-OBs werden in numerischer Reihenfolge ausgeführt.

Die Zeitverzögerungsereignisse ermöglichen Ihnen, die Ausführung eines Alarm-OB nach Ablauf einer vorgegebenen Zeitverzögerung einzurichten. Die Verzögerungszeit wird mit der Anweisung SRT\_DINT angegeben. Die Zeitverzögerungsereignisse unterbrechen den Programmzyklus, um den Zeitverzögerungs-OB auszuführen. Einem Zeitverzögerungsereignis darf nur ein Zeitverzögerungs-OB zugeordnet werden. Die CPU unterstützt vier Zeitverzögerungsereignisse.

Die Prozessalarmereignisse werden durch eine Veränderung in der Hardware ausgelöst, z.B. eine steigende oder fallende Flanke an einem Eingang oder ein HSC-Ereignis (schneller Zähler). Für jedes Prozessalarmereignis kann ein Alarm-OB ausgewählt werden. Die Hardware-Ereignisse werden in der Gerätekonfiguration aktiviert. Die OBs für das Ereignis werden in der Gerätekonfiguration oder mit einer Anweisung ATTACH im Anwenderprogramm angegeben. Die CPU unterstützt mehrere Prozessalarmereignisse. Die genaue Anzahl der Ereignisse richtet sich nach der Variante der CPU und der Anzahl der Eingänge.

Die Zeit- und Diagnosefehlerereignisse werden ausgelöst, wenn die CPU einen Fehler erkennt. Diese Ereignisse sind in einer Klasse mit höherer Priorität als die anderen Alarmereignisse und können die Ausführung der Zeitverzögerungs-, Weckalarm- und Prozessalarmereignisse unterbrechen. Für jedes Zeitfehler- und Diagnosefehlerereignis kann ein Alarm-OB angegeben werden.

### Wissenswertes zu Prioritäten und Warteschlange für die Ausführung von Ereignissen

Die Zahl anstehender Ereignisse aus einer einzigen Quelle kann begrenzt werden, indem jedem Ereignistyp eine eigene Warteschlange zugewiesen wird. Sobald die maximale Zahl anstehender Ereignisse eines bestimmten Typs erreicht ist, wird das nächste Ereignis nicht mehr bearbeitet und geht verloren. Weitere Informationen zum Warteschlangenüberlauf finden Sie im folgenden Abschnitt "Wissenswertes zu Zeitfehlerereignissen".

Jedes Ereignis einer CPU hat eine Priorität. Sie können die Priorität eines OBs nicht ändern. Die Ereignisse werden im Allgemeinen in der Reihenfolge ihrer Priorität (höchste zuerst) bearbeitet. Ereignisse mit gleicher Priorität werden nach dem First-In-First-Out-Prinzip bearbeitet.

Tabelle 4- 2 OB-Ereignisse

Ereignis	OB-Nummer	Zulässige Anzahl	Startereignis	OB-Priorität
Programmzyklus	OB 1, OB 200 bis OB 65535	1 Programmzyklusereignis Mehrere OBs zulässig	<ul style="list-style-type: none"> <li>Anlauf-OB endet</li> <li>Letzter Programmzyklus-OB endet</li> </ul>	1
Anlauf	OB 100, OB 200 bis OB 65535	1 Anlaufereignis <sup>1, 2</sup> Mehrere OBs zulässig	Wechsel von STOP nach RUN	1
Zeit	OB 200 bis OB 65535	Bis zu 4 Zeitereignisse <sup>3</sup> 1 OB je Ereignis	Verzögerungs-OB-Ereignis ist geplant	3
			Zyklus-OB-Ereignis ist geplant	7
Prozess	OB 200 bis OB 65535	Bis zu 50 Prozessereignisse <sup>4</sup> 1 OB je Ereignis	Flanken: <ul style="list-style-type: none"> <li>Ereignisse steigende Flanke: max. 16</li> <li>Ereignisse fallende Flanke: max. 16</li> </ul>	5
			Bei HSC: <ul style="list-style-type: none"> <li>CV=PV: max. 6</li> <li>Richtungswechsel: max. 6</li> <li>Externes Rücksetzen: max. 6</li> </ul>	6

Ereignis	OB-Nummer	Zulässige Anzahl	Startereignis	OB-Priorität
Diagnosefehler	OB 82	1 Ereignis (nur, wenn OB 82 geladen war)	Modul sendet einen Fehler	9
Zeitfehler	OB 80	1 Ereignis (nur, wenn OB 80 geladen war) <sup>5</sup>	<ul style="list-style-type: none"> <li>• Maximale Zykluszeit wurde überschritten</li> <li>• Ein zweiter Alarm (Weck- oder Verzögerungsalarm) wurde gestartet, bevor die CPU die Ausführung des ersten Alarms beendet hat</li> </ul>	26

- <sup>1</sup> Das Anlauf- und das Programmzyklus-Ereignis treten nie gleichzeitig ein, weil der Anlauf zuerst beendet sein muss, bevor der Programmzyklus gestartet wird (Steuerung durch das Betriebssystem).
- <sup>2</sup> Nur das Diagnosefehlerereignis (OB 82) unterbricht das Anlaufereignis. Alle anderen Ereignisse werden für die Bearbeitung nach dem Anlaufereignis in die Warteschlange gestellt.
- <sup>3</sup> Die CPU stellt insgesamt 4 Zeitereignisse zur Verfügung, die gemeinsam von den Verzögerungs-OBs und den Zyklus-OBs genutzt werden. Die Anzahl der Verzögerungs- und Zyklus-OBs im Anwenderprogramm darf nicht größer als 4 sein.
- <sup>4</sup> Bei Verwendung der Anweisungen DETACH und ATTACH sind mehr als 50 Prozessereignisse möglich.
- <sup>5</sup> Sie können die CPU so konfigurieren, dass sie in RUN bleibt, wenn die maximale Zykluszeit überschritten wurde, oder Sie können mit der Anweisung RE\_TRIGR die Zykluszeit zurücksetzen. Wenn die maximale Zykluszeit jedoch in einem Zyklus zum zweiten Mal überschritten wird, geht die CPU in den Betriebszustand STOP.

Nachdem die Ausführung eines OBs mit einer Priorität von 2 bis 25 gestartet ist, kann die Bearbeitung dieses OBs nicht durch Auftreten eines weiteren Ereignisses unterbrochen werden. Ausnahme ist OB 80 (Zeitfehlerereignis mit der Priorität 26). Alle anderen Ereignisse werden für die spätere Bearbeitung in die Warteschlange gestellt, damit der aktuelle OB beendet werden kann.

## Latenzzeit

Die Ereignis-Latenzzeit (d.h. die Zeit zwischen der Mitteilung der CPU über das Auftreten eines Ereignisses und dem Start der Ausführung der ersten Anweisung im OB für die Ereignisbearbeitung) beträgt ca. 175 µs, wenn zum Zeitpunkt des Alarmereignisses nur ein Programmzyklus-OB als Bearbeitungsroutine aktiv ist.

## Wissenswertes zu Zeitfehlerereignissen

Das Auftreten eines von verschiedenen Zeitfehlern führt zu einem Zeitfehlerereignis. Die folgenden Zeitfehler werden unterstützt:

- Maximale Zykluszeit überschritten
- Gewünschter OB kann nicht gestartet werden
- Warteschlangenüberlauf

Der Fehler "Maximale Zykluszeit überschritten" tritt auf, wenn der Programmzyklus nicht innerhalb der angegebenen maximalen Zykluszeit beendet wird. Weitere Informationen zum Fehler "Maximale Zykluszeit überschritten", zum Konfigurieren der maximalen Zykluszeit und zum Zurücksetzen der Zykluszeit finden Sie im Abschnitt "Überwachen der Zykluszeit" im S7-1200 Systemhandbuch (Seite 87).

Der Fehler "Gewünschter OB kann nicht gestartet werden" tritt auf, wenn ein OB von einem Weckalarm, einem Verzögerungsalarm oder einem Uhrzeitalarm angefordert wird, doch bereits ausgeführt wird.

Der Fehler "Warteschlangenüberlauf" tritt auf, wenn die Alarmergebnisse schneller auftreten als sie verarbeitet werden können. Die Zahl anstehender Ereignisse kann begrenzt werden, indem jedem Ereignistyp eine eigene Warteschlange zugewiesen wird. Tritt ein Ereignis auf, wenn die entsprechende Warteschlange voll ist, wird ein Zeitfehlerereignis erzeugt.

Alle Zeitfehlerereignisse lösen die Ausführung von OB 80 (sofern vorhanden) aus. Wenn das Anwenderprogramm keinen OB 80 enthält, legt die Gerätekonfiguration der CPU die Reaktion der CPU auf den Zeitfehler fest:

- Bei der Standardkonfiguration für Zeitfehler, wenn z. B. ein zweiter Weckalarm gestartet wird, bevor die CPU die Ausführung des ersten beendet hat, bleibt die CPU in RUN.
- Bei der Standardkonfiguration für das Überschreiten der maximalen Zykluszeit wird die CPU in STOP versetzt.

Sie können die maximale Zykluszeit mit der Anweisung RE\_TRIGR zurücksetzen. Wenn jedoch der Fehler "Maximale Zykluszeit überschritten" zwei Mal in demselben Programmzyklus auftritt, ohne dass die Zykluszeit zurückgesetzt wird, geht die CPU in STOP, unabhängig davon, ob OB 80 vorhanden ist. Weitere Informationen finden Sie im Abschnitt "Überwachen der Zykluszeit" im S7-1200 Systemhandbuch (Seite 87).

OB 80 enthält Anlaufinformationen, anhand deren Sie ermitteln können, welches Ereignis und welcher OB den Zeitfehler erzeugt hat. Sie können in OB 80 Anweisungen programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen.

Tabelle 4-3 Anlaufinformationen für OB 80

Eingang	Datentyp	Beschreibung
fault_id	BYTE	16#01 - Maximale Zykluszeit überschritten 16#02 - Gewünschter OB kann nicht gestartet werden 16#07 und 16#09 - Warteschlangenüberlauf
csg_OBnr	OB_ANY	Nummer des OBs, der bei Auftreten des Fehlers ausgeführt wurde
csg_prio	UINT	Priorität des fehlerverursachenden OBs

Wenn Sie ein neues Projekt anlegen, ist kein Zeitfehler-OB 80 vorhanden. Wenn Sie möchten, fügen Sie einen Zeitfehler-OB 80 in Ihr Projekt ein. Hierfür doppelklicken Sie in der Projektnavigation unter "Programmbausteine" auf "Neuen Baustein hinzufügen", dann wählen Sie "Organisationsbaustein" und "Zeitfehler".

### Wissenswertes zu Diagnosefehlerereignissen

Analoge (lokale), PROFINET- und PROFIBUS-Geräte können Diagnosefehler erkennen und melden. Das Auftreten bzw. Verschwinden eines von verschiedenen Diagnosefehlern führt zu einem Diagnosefehlerereignis. Die folgenden Diagnosefehler werden unterstützt:

- Keine Anwenderspannung
- Oberer Grenzwert überschritten
- Unterer Grenzwert überschritten
- Drahtbruch
- Kurzschluss

Diagnosefehlerereignisse lösen die Ausführung von OB 82 (sofern vorhanden) aus. Ist OB 82 nicht vorhanden, ignoriert die CPU den Fehler. Wenn Sie ein neues Projekt anlegen, ist kein Diagnosefehler-OB 82 vorhanden. Wenn Sie möchten, fügen Sie einen Diagnosefehler-OB 82 in Ihr Projekt ein. Hierfür doppelklicken Sie in der Projektnavigation unter "Programmbausteine" auf "Neuen Baustein hinzufügen", dann wählen Sie "Organisationsbaustein" und "Diagnosefehler".

---

#### Hinweis

#### Diagnosefehler bei mehrkanaligen lokalen Analoggeräten (E/A, RTD und Thermoelement)

Der Diagnosefehleralarm von OB 82 kann nur die Diagnosefehler jeweils eines Kanals melden.

Wenn in zwei Kanälen eines mehrkanaligen Geräts Fehler auftreten, löst der zweite Fehler den OB 82 nur unter folgenden Bedingungen aus: Der Fehler des ersten Kanals wird behoben, die vom ersten Fehler ausgelöste Ausführung von OB 82 ist beendet und der zweite Fehler liegt weiterhin vor.

---

OB 82 enthält Anlaufinformationen, anhand deren Sie ermitteln können, ob das Ereignis wegen des Auftretens oder Verschwindens eines Fehlers ausgelöst wurde, und welches Gerät und welcher Kanal den Fehler gemeldet haben. Sie können in OB 82 Anweisungen programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen.

Tabelle 4-4 Anlaufinformationen für OB 82

Eingang	Datentyp	Beschreibung
IOstate	WORD	E/A-Zustand des Geräts: <ul style="list-style-type: none"> <li>• Bit 0 = 1, wenn die Konfiguration korrekt ist, und Bit 0 = 0, wenn die Konfiguration nicht mehr korrekt ist.</li> <li>• Bit 4 = 1, wenn ein Fehler vorliegt (Beispiel: Drahtbruch). (Bit 4 = 0, wenn kein Fehler vorliegt.)</li> <li>• Bit 5 = 1, wenn die Konfiguration <b>nicht</b> korrekt ist, und Bit 5 = 0, wenn die Konfiguration wieder korrekt ist.</li> <li>• Bit 6 = 1, wenn ein E/A-Zugriffsfehler aufgetreten ist. Die Hardwarekennung der E/A mit dem Zugriffsfehler finden Sie in laddr. (Bit 6 = 0, wenn kein Fehler vorliegt.)</li> </ul>
laddr	HW_ANY	Hardwarekennung des Geräts oder der Funktionseinheit, das bzw. die den Fehler gemeldet hat <sup>1</sup>
channel	UINT	Kanalnummer
multierror	BOOL	WAHR, wenn mehrere Fehler vorliegen

<sup>1</sup> Der Eingang laddr enthält die Hardwarekennung des Geräts bzw. der Funktionseinheit, der bzw. die den Fehler ausgegeben hat. Die Hardwarekennung wird automatisch zugewiesen, wenn Komponenten in die Geräte- oder Netzsicht eingefügt werden. Sie wird im Register "Konstanten" von PLC-Variablen angezeigt. Der Hardwarekennung wird zudem automatisch ein Name zugewiesen. Diese Einträge im Register "Konstanten" der PLC-Variablen können nicht geändert werden.

### 4.1.5 Überwachen der Zykluszeit

Die Zykluszeit ist die Zeit, die das Betriebssystem der CPU benötigt, um die zyklische Phase des Betriebszustands RUN auszuführen. Die CPU bietet zwei Verfahren zum Überwachen der Zykluszeit:

- Maximale Zykluszeit
- Feste Mindestzykluszeit

Die Zyklusüberwachung beginnt nach abgeschlossenem Anlaufereignis. Projektiert werden kann diese Funktion in der CPU unter "Gerätekonfiguration > Zykluszeit".

Die CPU überwacht den Zyklus ständig und reagiert, wenn die maximale Zykluszeit überschritten wird. Wird die eingestellte maximale Zykluszeit überschritten, so wird ein Fehler erzeugt und auf eine der beiden folgenden Arten behandelt:

- Wenn das Anwenderprogramm keinen OB 80 enthält, generiert die CPU einen Fehler und geht in STOP. (Sie können die Konfiguration der CPU ändern, um diesen Zeitfehler zu ignorieren und in RUN zu bleiben. Bei der Standardkonfiguration geht die CPU in STOP.)
- Wenn das Anwenderprogramm einen OB 80 enthält, führt die CPU den OB 80 aus.

4.1 Ausführung des Anwenderprogramms

Die Anweisung RE\_TRIGR (Zykluszeitüberwachung neu starten) ermöglicht das Rücksetzen des Zeitgebers für die Messung der Zykluszeit. Die Anweisung funktioniert jedoch nur, wenn sie in einem Programmzyklus-OB ausgeführt wird. Wird die Anweisung RE\_TRIGR in OB 80 ausgeführt, wird sie ignoriert. Wenn die maximale Zykluszeit in einem Programmzyklus zwei Mal überschritten wird und dabei keine Anweisung RE\_TRIGR zwischen den beiden Zeitüberschreitungen ausgeführt wird, geht die CPU sofort in STOP. Die wiederholte Ausführung der Anweisung RE\_TRIGR kann eine Endlosschleife oder einen sehr langen Zyklus verursachen.

Typischerweise wird der Zyklus so schnell wie möglich ausgeführt und der nächste Zyklus beginnt, sobald der vorherige beendet ist. Je nach Anwenderprogramm und Kommunikationsaufgaben kann die Zykluszeit jedoch variieren. Um solche Schwankungen zu vermeiden, unterstützt die CPU eine optionale feste Mindestzykluszeit (wird auch als "fester Zyklus" bezeichnet). Ist diese Funktion aktiviert und eine feste Mindestzykluszeit in ms eingegeben, so hält die CPU diese Mindestzykluszeit mit einer Toleranz von ± 1 ms für jeden Zyklus ein.

Führt die CPU einen Zyklus schneller aus als mit der Mindestzykluszeit festgelegt ist, so nutzt die CPU die verbleibende Zeit für Laufzeitdiagnosen und/oder für die Bearbeitung von Kommunikationsanforderungen. Auf diese Weise belegt die CPU immer für jeden Zyklus eine feste Zeit.

Wird der Zyklus nicht innerhalb der vorgegebenen Mindestzykluszeit beendet, so wird er normal bis zum Ende ausgeführt (einschließlich der Kommunikationsbearbeitung) und die Überschreitung der Mindestzeit erzeugt keine Systemreaktion. In der folgenden Tabelle sind die Bereiche und Voreinstellungen für die Zykluszeitüberwachung angegeben.

Tabelle 4- 5 Bereich für die Zykluszeit

Zykluszeit	Bereich (ms)	Voreinstellung
Maximale Zykluszeit <sup>1</sup>	1 bis 6000	150 ms
Feste Mindestzykluszeit <sup>2</sup>	1 bis maximale Zykluszeit	Deaktiviert

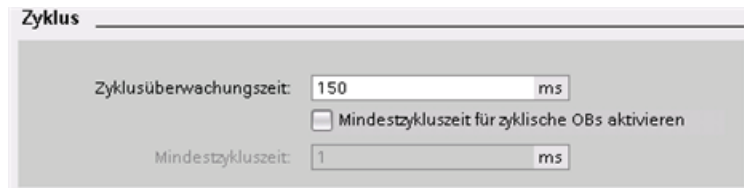
- <sup>1</sup> Die maximale Zykluszeit ist immer aktiviert. Richten Sie eine Zykluszeit zwischen 1 ms und 6000 ms ein. Standard sind 150 ms.
- <sup>2</sup> Die feste Mindestzykluszeit ist optional und standardmäßig deaktiviert. Richten Sie ggf. eine Zykluszeit zwischen 1 ms und der maximalen Zykluszeit ein.



## Konfigurieren von Zykluszeit und Kommunikationslast

In den CPU-Eigenschaften der Gerätekonfiguration können Sie die folgenden Parameter einstellen:

- **Zykluszeit:** Hier können Sie eine maximale Zykluszeit eingeben. Sie können auch eine feste Mindestzykluszeit definieren.



- **Kommunikationslast:** Sie können einen prozentualen Anteil der Zeit für Kommunikationsaufgaben festlegen.



Ausführliche Informationen über den Zyklus finden Sie unter "Überwachung des Zyklus" (Seite 87).

## 4.1.6 CPU-Speicher

### Speicherverwaltung

Die CPU stellt die folgenden Speicherbereiche für Anwenderprogramm, Daten und Konfiguration bereit:

- Der Ladespeicher ist ein nicht-flüchtiger Speicher für Anwenderprogramm, Daten und Konfiguration. Beim Laden eines Projekts in die CPU wird das Projekt zunächst im Ladespeicher abgelegt. Dieser Speicher befindet sich entweder auf einer Memory Card (sofern vorhanden) oder in der CPU. Der nicht-flüchtige Ladespeicher bleibt auch bei einem Spannungsausfall erhalten. Die Memory Card unterstützt einen größeren Speicherbereich als den in der CPU integrierten.
- Der Arbeitsspeicher ist ein flüchtiger Speicher für einige Elemente des Anwenderprojekts während der Bearbeitung des Anwenderprogramms. Die CPU kopiert einige Elemente des Projekts vom Ladespeicher in den Arbeitsspeicher. Dieser flüchtige Speicherbereich geht bei Spannungsausfall verloren und wird bei Spannungsrückkehr von der CPU wiederhergestellt.

- Der remanente Speicher ist ein nicht-flüchtiger Speicher für eine begrenzte Menge an Arbeitsspeicherwerten. Der remanente Speicherbereich dient zum Speichern der Werte ausgewählter Adressen des Anwenderspeichers bei Spannungsausfall. Wenn eine Abschaltung oder ein Spannungsausfall auftritt, stellt die CPU diese remanenten Werte beim Anlauf wieder her.

Um die Speicherauslastung des aktuellen Projekts anzuzeigen, klicken Sie mit der rechten Maustaste auf die CPU (oder einen ihrer Bausteine) und wählen im Kontextmenü den Befehl "Speicherauslastung". Um die Speicherauslastung der aktuellen CPU anzuzeigen, doppelklicken Sie auf "Online & Diagnose", erweitern "Diagnose" und wählen "Speicher".

## Remanenter Speicher

Datenverlust nach Spannungsausfall kann dadurch vermieden werden, dass bestimmte Daten als remanent definiert werden. Die folgenden Daten können als remanent konfiguriert werden:

- **Merker (M):** Sie können die genaue Breite des Speichers für Merker in der PLC-Variablen-tabelle oder in der Zuweisungsliste definieren. Der remanente Merkerspeicher beginnt immer an MB0 und läuft über die angegebene Anzahl von Bytes ununterbrochen weiter. Geben Sie diesen Wert in der PLC-Variablen-tabelle oder in der Zuweisungsliste an, indem Sie in der Funktionsleiste auf das Symbol "Remanent" klicken. Geben Sie die Anzahl der Merkerbytes an, die ab MB0 remanent sein sollen.
- **Variablen eines Funktionsbausteins (FB):** Wenn ein FB mit aktivierter Option "Optimiert" erstellt wurde, enthält der Schnittstelleneditor für diesen FB eine Spalte "Remanent". In dieser Spalte können Sie für jede Variable einzeln entweder "Remanent", "Nicht remanent" oder "Im IDB setzen" auswählen. Ein Instanz-DB, der beim Einfügen des FBs in den Programmiereditor angelegt wurde, weist ebenfalls diese Spalte auf. Sie können den remanenten Zustand einer Variable im Schnittstelleneditor des Instanz-DBs nur ändern, wenn Sie im optimierten FB im Bereich "Remanent" für die Variable die Option "Im IDB setzen" (Im Instanzdatenbaustein setzen) ausgewählt haben.

Wenn ein FB mit aktivierter Option "Standard - kompatibel mit S7-300/400" erstellt wurde, enthält der Schnittstelleneditor für diesen FB keine Spalte "Remanent". Ein Instanz-DB, der beim Einfügen des FBs in den Programmiereditor angelegt wurde, weist eine Spalte "Remanent" auf, die bearbeitet werden kann. In diesem Fall wirkt sich die Auswahl der Option "Remanent" für eine Variable so aus, dass **alle** Variablen ausgewählt werden. Genauso wirkt es sich aus, wenn Sie die Option für eine Variable deaktivieren. Dann werden **alle** Variablen deaktiviert. Bei einem FB, für den die Option "Standard - kompatibel mit S7-300/400" aktiviert ist, können Sie den remanenten Zustand im Editor des Instanz-DBs ändern, doch es werden alle Variablen gleichzeitig auf denselben Zustand gesetzt.

Nachdem Sie den FB angelegt haben, können Sie die Option "Standard - kompatibel mit S7-300/400" nicht mehr ändern. Sie können diese Option nur beim Anlegen des FBs auswählen. Um zu ermitteln, ob ein vorhandener FB mit aktivierter Option "Optimiert" oder "Standard - kompatibel mit S7-300/400" angelegt wurde, klicken Sie in der Projektnavigation mit der rechten Maustaste auf den FB, wählen "Eigenschaften" und dann "Attribute". Ist das Kontrollkästchen "Optimiert" aktiviert, wurde der Baustein mit optimiertem Bausteinzugriff angelegt. Andernfalls ist es ein Standardbaustein und kompatibel mit den S7-300/400 CPUs.

- **Variablen eines globalen Datenbausteins:** Das Verhalten eines globalen DBs in Bezug auf die Zuweisung des remanenten Zustands ähnelt dem eines FBs. Je nach Einstellung des Bausteinzugriffs können Sie den remanenten Zustand entweder für einzelne oder für alle Variablen eines globalen Datenbausteins definieren.
  - Wenn Sie beim Anlegen des DBs "Optimiert" auswählen, können Sie den remanenten Zustand für jede Variable einzeln festlegen.
  - Wenn Sie beim Anlegen des DBs "Standard - kompatibel mit S7-300/400" auswählen, gilt die Einstellung des remanenten Zustands für alle Variablen des DBs, d. h., es sind entweder alle Variablen remanent oder es ist keine Variable remanent.

Insgesamt können 10240 Bytes an Daten remanent sein. Um zu ermitteln, wieviele Bytes verfügbar sind, klicken Sie in der PLC-Variablen-tabelle oder in der Zuweisungsliste in der Funktionsleiste auf das Symbol "Remanent". Hier geben Sie zwar den remanenten Bereich für den Merkerspeicher an, doch die zweite Zeile gibt den verbleibenden Gesamtspeicher für M und DB zusammen an. Beachten Sie, dass Sie alle Datenbausteine mit remanenten Variablen übersetzen müssen, damit dieser Wert korrekt ist.

#### 4.1.6.1 System- und Taktmerker

In den CPU-Eigenschaften können Sie Bytes für "Systemspeicher" und "Taktmerker" aktivieren. Die einzelnen Bits dieser Funktionen können in Ihrer Programmlogik über die Variablennamen referenziert werden.

- Sie können im Merkerbereich ein Byte als Systemspeicher zuweisen. Das Byte des Systemmerkers bietet die folgenden vier Bits, die von Ihrem Anwenderprogramm über die folgenden Variablennamen referenziert werden können:
  - Erster Zyklus: Das Bit (Variablenname "FirstScan") wird nach dem Ende des Anlauf-OBs für den ersten Zyklus auf 1 gesetzt. (Nach dem Ende des ersten Zyklus wird das Bit "Erster Zyklus" auf 0 gesetzt.)
  - Das Bit "Diagnosezustand geändert" (Variablenname "DiagStatusUpdate") wird einen Zyklus lang auf 1 gesetzt, nachdem die CPU ein Diagnoseereignis erfasst hat. Weil die CPU das Bit "Diagnosediagramm geändert" erst am Ende der ersten Ausführung des Programmzyklus-OBs setzt, kann Ihr Anwenderprogramm nicht erkennen, ob während der Ausführung des Anlauf-OBs oder während der ersten Ausführung des Programmzyklus-OBs eine Diagnoseänderung auftrat.
  - Immer 1 (high): Das Bit (Variablenname "AlwaysTRUE") ist immer auf 1.
  - Immer 0 (low): Das Bit (Variablenname "AlwaysFALSE") ist immer auf 0.
- Sie können im Merkerbereich ein Byte als Taktmerker zuweisen. Jedes Bit dieses als Taktmerker konfigurierten Bytes erzeugt einen Rechteckimpuls. Das Byte des Taktmerkers bietet 8 verschiedene Frequenzen, von 0,5 Hz (langsam) bis 10 Hz (schnell). Sie können diese Bits, besonders in Verbindung mit Flankenweisungen, als Steuerbits für die zyklische Auslösung von Aktionen im Anwenderprogramm verwenden.

Die CPU initialisiert diese Bytes beim Wechsel von STOP in STARTUP. Die Bits des Taktmerkers wechseln während der Betriebsarten STARTUP und RUN synchron zum CPU-Takt.

#### VORSICHT

Ein Überschreiben der Bits von Systemspeicher oder Taktmerker kann die Daten in diesen Funktionen beschädigen und bewirken, dass Ihr Anwenderprogramm fehlerhaft arbeitet, was zu Sachschaden und Verletzungen des Personals führen kann.

Weil Taktmerker und Systemspeicher nicht reservierter Speicher im Bereich der Merker sind, können Anweisungen und Kommunikation in diese Adressen schreiben und Daten beschädigen.

Vermeiden Sie das Schreiben von Daten in diese Adressen, um sicherzustellen, dass diese Funktionen einwandfrei arbeiten, und implementieren Sie immer eine Not-Aus-Schaltung für Ihren Prozess oder Ihre Maschine.

Der Systemspeicher konfiguriert ein Byte mit Bits, die bei einem spezifischen Ereignis eingeschaltet werden (Wert = 1).

**Systemmerkerbits**

Verwendung des Systemmerkerbytes aktivieren

Adresse des Systemmerkerbytes (MBx):

Erster Zyklus:

Diagnosestatus geändert:

Immer 1 (high):

Immer 0 (low):

Tabelle 4- 6 Systemspeicher

7	6	5	4	3	2	1	0
Reserviert Wert 0				Immer ausgeschaltet Wert 0	Immer eingeschaltet Wert 1	Diagnosestatusanzeige <ul style="list-style-type: none"> <li>• 1: Änderung</li> <li>• 0: Keine Änderung</li> </ul>	Anzeige erster Zyklus <ul style="list-style-type: none"> <li>• 1: Erster Zyklus nach Anlauf</li> <li>• 0: Nicht der erste Zyklus</li> </ul>

Der Taktmerker konfiguriert ein Byte, das die einzelnen Bits in bestimmten Abständen ein- und ausschaltet. Jeder Taktmerker erzeugt einen Rechteckimpuls im entsprechenden Merkerspeicher M. Diese Bits können, vor allem in Verbindung mit Flankenweisungen, als Steuerbits für die zyklische Auslösung von Aktionen im Anwendercode verwendet werden.

**Taktmerkerbits**

Verwendung des Taktmerkerbytes aktivieren

Adresse des Taktmerkerbytes (MBx):

Takt 10 Hz:

Takt 5 Hz:

Takt 2,5 Hz:

Takt 2 Hz:

Takt 1,25 Hz:

Takt 1 Hz:

Takt 0,625 Hz:

Takt 0,5 Hz:

Tabelle 4- 7 Taktmerker

Bitnummer	7	6	5	4	3	2	1	0
Variablenname								
Period(en)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Frequenz (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

Weil der Taktmerker asynchron zum CPU-Zyklus läuft, kann sich der Zustand des Taktmerkers während eines langen Zyklus mehrere Male ändern.

### 4.1.7 Diagnosepuffer

Die CPU unterstützt einen Diagnosepuffer, der für jedes Diagnoseereignis einen Eintrag enthält. Jeder Eintrag umfasst das Datum und die Uhrzeit, zu denen das Ereignis aufgetreten ist, eine Ereigniskategorie und eine Ereignisbeschreibung. Die Einträge werden in chronologischer Reihenfolge angezeigt, wobei das jüngste Ereignis an oberster Stelle steht. In diesem Puffer werden bis zu 50 aktuelle Ereignisse gespeichert. Ist der Puffer voll, wird immer das älteste Ereignis durch ein neues Ereignis überschrieben. Bei einer Unterbrechung der Stromversorgung werden die Ereignisse gespeichert.

Im Diagnosepuffer werden die folgenden Ereignistypen erfasst:

- Jedes Diagnoseereignis des Systems, CPU-Fehler und Modulfehler
- Jeder Zustandswechsel der CPU (Anlauf, Wechsel in Betriebszustand STOP oder RUN)

Für den Zugriff auf den Diagnosepuffer (Seite 724) müssen Sie online sein. Den Puffer finden Sie unter "Online & Diagnose > Diagnose > Diagnosepuffer".

### 4.1.8 Echtzeituhr

Die CPU verfügt über eine Echtzeituhr. Wenn die CPU ausgeschaltet ist, wird die Uhr über einen Hochleistungskondensator mit Spannung versorgt. Der Hochleistungskondensator wird aufgeladen, wenn die CPU eingeschaltet ist. War die CPU mindestens 24 Stunden eingeschaltet, so reicht die Ladung des Hochleistungskondensators für den Betrieb der Uhr typischerweise für 20 Tage.

STEP 7 stellt für die Echtzeituhr die Systemzeit ein, die einen werkseitig voreingestellten Standardwert hat bzw. den Standardwert durch Rücksetzen auf die Werkseinstellungen erhält. Um die Echtzeituhr zu nutzen, müssen Sie sie einstellen. Zeitstempel wie die von Diagnosepuffereinträgen, Datenprotokolldateien und Datenprotokolleinträgen beruhen auf der Systemzeit. Sie stellen die Echtzeituhr über die Funktion "Uhrzeit einstellen" (Seite 721) in der Ansicht "Online & Diagnose" der Online-CPU ein. STEP 7 berechnet dann die Systemzeit anhand der von Ihnen eingestellten Zeit plus oder minus dem Zeitunterschied des Windows-Betriebssystems zur UTC (Coordinated Universal Time). Wird für die Echtzeituhr die Ortszeit eingestellt, erzeugt dies eine Systemzeit, die der UTC entspricht, sofern die Einstellungen für Zeitzone und Sommer-/Winterzeitschaltung Ihres Windows-Betriebssystems Ihrem Gebietsschema entsprechen.

STEP 7 enthält Anweisungen (Seite 263) zum Lesen und Schreiben der Systemzeit (RD\_SYS\_T und WR\_SYS\_T), zum Lesen der Ortszeit (RD\_LOC\_T) und zum Einstellen der Zeitzone (SET\_TIMEZONE). Die Anweisung RD\_LOC\_T berechnet die Ortszeit anhand der Zeitzone und der Zeitverschiebung für die Sommer-/Winterzeit, die Sie in der Konfiguration der Echtzeituhr in den allgemeinen Eigenschaften der CPU (Seite 133) eingegeben haben. Über diese Einstellungen können Sie die Zeitzone für Ihre Ortszeit einrichten, Sie können optional die Sommerzeitumschaltung aktivieren und die Anfangs- und Endezeiten für die Sommer-/Winterzeitumschaltung eingeben. Diese Einstellungen können Sie auch mit den Anweisungen SET\_TIMEZONE konfigurieren.

#### 4.1.9 Konfigurieren der Ausgänge für den Wechsel von RUN in STOP

Sie können das Verhalten der digitalen und analogen Ausgänge für den Betriebszustand STOP der CPU konfigurieren. Für jeden Ausgang einer CPU, eines SB oder SM können Sie die Werte der Ausgänge einfrieren oder einen Ersatzwert aufschalten:

- Durch einen vorgegebenen Ausgangswert ersetzen (Voreinstellung): Sie geben für jeden Ausgang (Kanal) der CPU, des SB oder SM einen Ersatzwert ein.

Der voreingestellte Ersatzwert für digitale Ausgänge ist AUS, für analoge Ausgänge ist er 0.

- Einfrieren der Ausgänge im letzten Zustand: Die Ausgänge behalten ihre aktuellen Werte, die sie zu der Zeit hatten, als der Wechsel von RUN nach STOP auftrat. Nach dem Anlauf werden die Ausgänge auf den Standardersatzwert gesetzt.

Sie konfigurieren das Verhalten der Ausgänge in der Gerätekonfiguration. Wählen Sie die einzelnen Geräte und öffnen Sie das Register "Eigenschaften", um die Ausgänge des jeweiligen Geräts zu konfigurieren.

Wenn die CPU von RUN nach STOP wechselt, speichert die CPU das Prozessabbild und schreibt die entsprechenden Werte für digitale und analoge Ausgänge entsprechend der Konfiguration.

## 4.2 Datenspeicher, Speicherbereiche, E/A und Adressierung

### 4.2.1 Zugriff auf Daten der S7-1200

STEP 7 vereinfacht die symbolische Programmierung. Dazu erstellen Sie für die Adressen der Daten symbolische Namen oder "Variablen", die entweder in Form von PLC-Variablen für Speicheradressen und E/A oder in Form von lokalen Variablen innerhalb eines Codebausteins vorkommen. Zum Einfügen dieser Variablen in Ihr Anwenderprogramm geben Sie einfach den Variablennamen für den gewünschten Anweisungsparameter ein.

Zur Verdeutlichung, wie die CPU Speicherbereiche strukturiert und adressiert, wird im Folgenden dargestellt, wie PLC-Variablen auf die "absolute" Adressierung der Daten verweisen. Die CPU bietet mehrere Möglichkeiten für die Datenspeicherung während der Ausführung des Anwenderprogramms:

- **Globaler Speicher:** Die CPU bietet eine Vielzahl von spezialisierten Speicherbereichen, einschließlich Eingänge (E), Ausgänge (A) und Merker (M). Dieser Speicher ist für alle Codebausteine ohne Einschränkung zugänglich.
- **PLC-Variablen-tabelle:** In der STEP 7-PLC-Variablen-tabelle können Sie für bestimmte Adressen im Speicher symbolische Namen eingeben. Dieser Variablen gelten global im gesamten STEP 7-Programm und gestatten die Programmierung mit aussagekräftigen Namen.
- **Datenbaustein (DB):** Sie können in Ihr Anwenderprogramm DBs zum Speichern von Daten für die Codebausteine einfügen. Die gespeicherten Daten bleiben nach der Ausführung des zugehörigen Codebausteins erhalten. In einem "globalen" DB werden Daten gespeichert, die von allen Codebausteinen verwendet werden können, in einem Instanz-DB werden jedoch nur Daten für einen bestimmten FB gespeichert, und er ist entsprechend der Parameter des FBs strukturiert.
- **Temporärer Speicher:** Bei jedem Aufruf eines Codebausteins gibt das Betriebssystem der CPU temporären bzw. lokalen Speicherplatz (L) frei, der bei der Ausführung des Bausteins genutzt werden kann. Ist die Ausführung des Codebausteins beendet, weist die CPU den lokalen Speicher für die Ausführung anderer Codebausteine zu.

Jeder Speicherplatz hat eine eindeutige Adresse. Anhand dieser Adresse kann Ihr Anwenderprogramm auf die Informationen an diesem Speicherplatz zugreifen. Durch Verweise auf Speicherbereiche für Eingänge (E) oder Ausgänge (A), z. B. E0.3 oder A1.7, erfolgt ein Zugriff auf das Prozessbild. Für den direkten Zugriff auf den physischen Eingang oder Ausgang fügen Sie den Verweis ":P" (z. B. E0.3:P, A1.7:P oder "Stop:P") als Anhang hinzu.

Tabelle 4- 8 Speicherbereiche

Speicherbereich	Beschreibung	Forcen	Remanent
E Prozessabbild der Eingänge	Wird zu Beginn des Zyklus aus den physischen Eingängen kopiert	Nein	Nein
E_:P <sup>1</sup> (physischer Eingang)	Direktes Lesen der physischen Eingänge von CPU, SB oder SM	Ja	Nein
A Prozessabbild der Ausgänge	Wird zu Beginn des Zyklus in die physischen Ausgänge kopiert	Nein	Nein
A_:P <sup>1</sup> (physischer Ausgang)	Direktes Schreiben in die physischen Ausgänge von CPU, SB oder SM	Ja	Nein
M Merker	Steuerung und Datenspeicher	Nein	Ja (optional)
L temporärer Speicher	Temporäre, lokale Daten für einen Baustein	Nein	Nein
DB Datenbaustein	Datenspeicher und auch Parameterspeicher für FBs	Nein	Ja (optional)

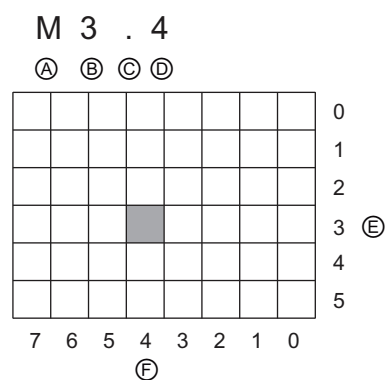
<sup>1</sup> Für den direkten Zugriff (Lesen oder Schreiben) auf die physischen Eingänge und Ausgänge hängen Sie den Code ":P" an die Adresse oder Variable an (z. B. E0.3:P, A1.7:P oder "Stop:P").



Jeder Speicherplatz hat eine eindeutige Adresse. Anhand dieser Adresse kann Ihr Anwenderprogramm auf die Informationen an diesem Speicherplatz zugreifen. Die absolute Adresse setzt sich aus den folgenden Elementen zusammen:

- Speicherbereichskennung (wie E, A oder M)
- Größe der Daten, auf die zugegriffen werden soll (wie "B" für Byte, "W" für Word oder "D" für DWord)
- Anfangsadresse der Daten (wie Byte 3 oder Wort 3)

Beim Zugriff auf ein Bit in der Adresse eines Booleschen Werts geben Sie keine Mnemonik für die Größe ein. Sie geben nur den Speicherbereich, die Byte-Adresse und die Bitadresse der Daten ein (wie E0.0, A0.1 oder M3.4).



- |   |                                  |   |                            |
|---|----------------------------------|---|----------------------------|
| A | Speicherbereichskennung          | E | Bytes des Speicherbereichs |
| B | Adresse des Byte: Byte 3         | F | Bits des ausgewählten Byte |
| C | Trennzeichen ("Byte.Bit")        |   |                            |
| D | Bitadresse im Byte (Bit 4 von 8) |   |                            |

In dem Beispiel folgt auf den Speicherbereich und die Adresse des Bytes (M = Bereich der Merker und 3 = Byte 3) ein Punkt ("."), um die Adresse des Bits (Bit 4) abzutrennen.

### Zugriff auf Daten in den Speicherbereichen der CPU

STEP 7 vereinfacht die symbolische Programmierung. Typischerweise werden Variablen entweder in PLC-Variablen, einem Datenbaustein oder in der Schnittstelle oberhalb von OB, FC oder FB angelegt. Diese Variablen umfassen einen Namen, Datentyp, Versatz und Kommentar. Außerdem kann in einem Datenbaustein ein Startwert angegeben werden. Diese Variablen können Sie während der Programmierung nutzen, indem Sie den Variablennamen als Parameter für die Anweisung eingeben. Optional können Sie den absoluten Operanden (Speicher, Bereich, Größe und Versatz) als Anweisungsparameter eingeben. Die Beispiele in den folgenden Abschnitten zeigen, wie Sie absolute Operanden eingeben. Das Zeichen % wird vom Programmiereditor automatisch vor dem absoluten Operanden eingefügt. Im Programmiereditor haben Sie die Auswahl zwischen folgenden Ansichten: symbolisch, symbolisch und absolut, absolut.

**E (Prozessabbild der Eingänge):** Die CPU fragt die Peripherieeingänge (physikalischen Eingänge) unmittelbar vor der Ausführung eines Zyklus-OB in jedem Zyklus ab und schreibt diese Werte in das Prozessabbild der Eingänge. Auf das Prozessabbild der Eingänge können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Es ist sowohl der Lese- als auch der Schreibzugriff erlaubt, jedoch werden die Eingänge des Prozessabbilds typischerweise nur gelesen.

Tabelle 4- 9 Absolute Adressierung für den Speicherbereich E

Bit	E[Byteadresse].[Bitadresse]	E0.1
Byte, Wort oder Doppelwort	E[Größe][Adresse des Anfangsbyte]	EB4, EW5 oder ED12

Wenn Sie den Verweis ":P" an die Adresse anhängen, können Sie die digitalen und analogen Eingänge von CPU, SB oder SM direkt lesen. Der Unterschied zwischen dem Zugriff über E\_:P statt über E liegt darin, dass die Daten direkt von den angesprochenen Eingängen kommen und nicht aus dem Prozessabbild der Eingänge. Dieser Zugriff über E\_:P wird auch als direkter Lesezugriff bezeichnet, weil die Daten direkt aus der Quelle statt aus einer Kopie, die bei der letzten Aktualisierung des Prozessabbilds der Eingänge erstellt wurde, gelesen werden.

Weil die physikalischen Eingänge ihre Werte direkt aus den mit ihnen verbundenen Feldgeräten erhalten, darf in diese Eingänge nicht geschrieben werden. Zugriffe über E\_:P sind somit reine Lesezugriffe, im Gegensatz zu Zugriffen auf Eingänge, die gelesen oder geschrieben werden können.

Zugriffe über E\_:P sind ferner durch die Größe der Eingänge begrenzt, die von CPU, SB oder SM unterstützt werden (gerundet auf das nächste Byte). Sind beispielsweise die Eingänge eines SB mit 2 DE/2 DA so konfiguriert, dass sie bei E4.0 beginnen, kann mit E4.0:P and E4.1:P oder EB4:P darauf zugegriffen werden. Der Zugriff auf E4.2:P bis E4.7:P wird nicht als Fehler erkannt, ist aber sinnlos, weil diese Adressen nicht belegt sind. Zugriffsversuche auf EW4:P und ED4:P sind verboten, weil sie den Byteversatz dieses SB überschreiten.

Zugriffe über E\_:P haben keinen Einfluss auf den im Prozessabbild der Eingänge gespeicherten Wert.

Tabelle 4- 10 Absolute Adressierung für den Speicherbereich E (direkt)

Bit	E[Byteadresse].[Bitadresse]:P	E0.1:P
Byte, Wort oder Doppelwort	E[Größe][Anfangsadresse des Byte]:P	EB4:P, EW5:P oder ED12:P

**A (Prozessabbild der Ausgänge):** Die CPU kopiert die im Prozessabbild der Ausgänge gespeicherten Werte in die physischen Ausgänge. Auf das Prozessabbild der Ausgänge können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Auf die Ausgänge des Prozessabbilds ist sowohl der Lese- als auch der Schreibzugriff erlaubt.

Tabelle 4- 11 Absolute Adressierung für den Speicherbereich A

Bit	A[Byteadresse].[Bitadresse]	A1.1
Byte, Wort oder Doppelwort	A[Größe][Adresse des Anfangsbyte]	AB5, AW10, AD40

Wenn Sie den Verweis ":P" an die Adresse anhängen, können Sie direkt in die digitalen und analogen Ausgänge von CPU, SB oder SM schreiben. Der Unterschied zwischen dem Zugriff über A\_:P statt über A liegt darin, dass die Daten direkt zu den angesprochenen Ausgängen gelangen und zusätzlich zum Prozessabbild der Ausgänge (es wird in beide Bereiche geschrieben). Dieser Zugriff über A\_:P wird auch als direkter Schreibzugriff bezeichnet, weil die Daten direkt an die Zieladresse gesendet werden und diese nicht auf die nächste Aktualisierung aus dem Prozessabbild der Ausgänge warten muss.

Weil die physikalischen Ausgänge direkt die mit ihnen verbundenen Feldgeräte steuern, dürfen diese Ausgänge nicht gelesen werden. Zugriffe über A\_:P sind somit reine Schreibzugriffe, im Gegensatz zu Zugriffen auf Ausgänge, die gelesen oder geschrieben werden können.

Zugriffe über A\_:P sind ferner durch die Größe der Ausgänge begrenzt, die von CPU, SB oder SM unterstützt werden (gerundet auf das nächste Byte). Sind beispielsweise die Ausgänge eines SB mit 2 DE/2 DA so konfiguriert, dass sie bei A4.0 beginnen, kann mit A4.0:P and A4.1:P oder AB4:P darauf zugegriffen werden. Der Zugriff auf A4.2:P bis A4.7:P wird nicht als Fehler erkannt, ist aber sinnlos, weil diese Adressen nicht belegt sind. Zugriffsversuche auf AW4:P und AD4:P sind verboten, weil sie den Byteversatz dieses SB überschreiten.

Zugriffe über A\_:P beeinflussen den physischen Ausgang sowie den im Prozessabbild der Ausgänge gespeicherten Wert.

Tabelle 4- 12 Absolute Adressierung für den Speicherbereich A (direkt)

Bit	A[Byteadresse].[Bitadresse]:P	A1.1:P
Byte, Wort oder Doppelwort	A[Größe][Anfangsadresse des Byte]:P	AB5:P, AW10:P oder AD40:P

**M (Speicherbereich der Merker):** Den Speicherbereich der Merker (M) können Sie für Steuerungsrelais und Daten verwenden, um Zwischenergebnisse von Anweisungen oder andere Steuerungsinformationen zu speichern. Auf den Speicherbereich der Merker können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Auf den Merkerspeicher ist sowohl der Lese- als auch der Schreibzugriff erlaubt.

Tabelle 4- 13 Absolute Adressierung für den Speicherbereich M

Bit	M[Byteadresse].[Bitadresse]	M26.7
Byte, Wort oder Doppelwort	M[Größe][Adresse des Anfangsbyte]	MB20, MW30, MD50

**Temp (temporärer Speicher):** Die CPU ordnet den temporären Speicher nach Bedarf zu. Die CPU ordnet dem Codebaustein den temporären Speicher zu der Zeit zu, zu der der Codebaustein gestartet wird (bei einem OB) bzw. aufgerufen wird (bei einer FC oder einem FB). Die Zuordnung des temporären Speichers für einen Codebaustein kann die gleichen Adressen des temporären Speichers wiederverwenden, die zuvor von einem anderen OB, FB oder einer FC verwendet wurden. Die CPU initialisiert den temporären Speicher nicht zu der Zeit der Zuordnung, deshalb kann der temporäre Speicher beliebige Werte enthalten.

Der temporäre Speicher ist dem Merkerspeicher (M) ähnlich. Es gibt jedoch einen wichtigen Unterschied: Der Merkerspeicher hat "globalen" Geltungsbereich, während der temporäre Speicher "lokalen" Geltungsbereich hat:

- Speicherbereich der Merker: Jeder OB, FB und jede FC kann auf die Daten im Speicherbereich der Merker zugreifen, d.h. die Daten stehen global für alle Elemente des Anwenderprogramms zur Verfügung.
- Temporärer Speicher: Der Zugriff auf die Daten im temporären Speicher ist auf den OB, FB oder die FC begrenzt, der bzw. die die Adresse im temporären Speicher erstellt bzw. deklariert hat. Adressen im temporären Speicher bleiben lokal und werden nicht von verschiedenen Codebausteinen gemeinsam genutzt, auch dann nicht, wenn der Codebaustein einen anderen Codebaustein aufruft. Beispiel: Wenn ein OB eine FC aufruft, kann die FC nicht auf den temporären Speicher des OBs, von dem sie aufgerufen wurde, zugreifen.

Die CPU stellt für jede der drei OB-Prioritätsklassen temporären (lokalen) Speicher zur Verfügung:

- 16 KB für Anlauf und Programmzyklus, einschließlich der zugehörigen FBs und FCs
- 4 KB für Standardalarmereignisse, einschließlich FBs und FCs
- 4 KB für Fehleralarmereignisse, einschließlich FBs und FCs

Auf den temporären Speicher können Sie nur über die symbolische Adressierung zugreifen.

**DB (Datenbaustein):** Nutzen Sie Datenbausteine zum Speichern verschiedener Arten von Daten, auch Zwischenergebnisse einer Anweisung oder andere Steuerungsparameter für FBs und Datenstrukturen für viele Anweisungen wie Zeiten und Zähler. Auf Datenbausteine können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Für nicht schreibgeschützte Datenbausteine ist sowohl der Lese- als auch der Schreibzugriff erlaubt. Für schreibgeschützte Datenbausteine ist nur der Lesezugriff erlaubt.

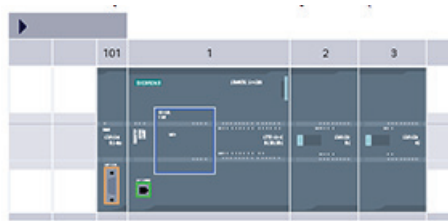
Tabelle 4- 14 Absolute Adressierung für den Speicherbereich DB

Bit	DB[Datenbausteinnummer].DBX[Byteadresse].[Bitadresse]	DB1.DBX2.3
Byte, Wort oder Doppelwort	DB[Datenbausteinnummer].DB[Größe].[Anfangsbyteadresse]	DB1.DBB4, DB10.DBW2, DB20.DBD8

**Hinweis**

Wenn Sie eine absolute Adresse angeben, stellt STEP 7 dieser Adresse das Zeichen "%" voran, um kenntlich zu machen, dass es sich um eine absolute Adresse handelt. Bei der Programmierung können Sie eine absolute Adresse entweder mit dem oder ohne das Zeichen "%" eingeben (Beispiel: %E0.0 oder E.0). Wenn Sie das Zeichen "%" weglassen, fügt STEP 7 es ein.

## E/A in der CPU und in E/A-Modulen konfigurieren



Geräteübersicht						
Baugruppe	Steckp.	E-Adresse	A-Adresse	Typ	Bestel	
	103					
	102					
RS485_1	101			CM 1241 (RS485)	6ES7	
PLC_1	1			CPU 1214C D0/D0C	6ES7	
DI14/DO10	1.1	0...1	0...1	DI14/DO10		
AI2	1.2	64...67		AI2		
AO1 x 12Bit	1.3		80...81	AO1 Signalboard	6ES7	
HSC_1	1.16	1000....		Schneller Zähler (F)		
HSC_2	1.17			Schneller Zähler (F)		
HSC_3	1.18			Schneller Zähler (F)		
HSC_4	1.19			Schneller Zähler (F)		
HSC_5	1.20			Schneller Zähler (F)		
HSC_6	1.21			Schneller Zähler (F)		
Pulse_1	1.32			Impulsgenerator (F)		
Pulse_2	1.33			Impulsgenerator (F)		
PROFINET... X1				PROFINET-Schnitts.		
DIB x DC24V...	2	8		SM 1221 DIB x DC	6ES7	

Wenn Sie eine CPU und E/A-Module in Ihren Konfigurationsbildschirm einfügen, werden E- und A-Adressen automatisch zugewiesen. Sie können die voreingestellte Adressierung ändern, indem Sie im Konfigurationsbildschirm das Adressfeld auswählen und neue Zahlen eingeben.

- Digitale Eingänge und Ausgänge werden in Gruppen zu 8 Punkten (1 Byte) zugewiesen, unabhängig davon, ob das Modul alle Ein- bzw. Ausgänge (Punkte) nutzt oder nicht.
- Analoge Eingänge und Ausgänge werden in Gruppen zu je 2 Ein- bzw. Ausgängen zugewiesen (4 Byte).

Das Bild zeigt ein Beispiel für eine CPU 1214C mit zwei SMs und einem SB. In diesem Beispiel können Sie in der Adresse des Moduls DE8 die 8 in 2 ändern. Das Werkzeug unterstützt Sie, indem Adressbereiche, die die falsche Größe haben oder mit anderen Adressen in Konflikt stehen, geändert werden.

## 4.3 Verarbeitung von Analogwerten

Analoge Signalmodule liefern Eingangssignale oder erwarten Ausgangswerte, die entweder einen Spannungsbereich oder einen Strombereich darstellen. Diese Bereiche sind  $\pm 10$  V,  $\pm 5$  V,  $\pm 2,5$  V oder 0 bis 20 mA. Die von den Modulen ausgegebenen Werte sind ganzzahlige Werte. Dabei stellt 0 bis 27648 den Nennbereich für Strom und -27648 bis 27648 den für Spannung dar. Werte außerhalb dieser Bereiche stellen entweder einen Überlauf oder einen Unterlauf dar. Einzelheiten hierzu finden Sie in den Tabellen Darstellung Analogeingang (Seite 818) und Darstellung Analogausgang (Seite 819).

In Ihrem Steuerungsprogramm müssen Sie diese Werte wahrscheinlich in physikalischen Einheiten verwenden, um beispielsweise Volumen, Temperatur, Gewicht oder einen anderen quantitativen Wert darzustellen. Hierfür müssen Sie den Analogeingang zunächst in eine Realzahl (Gleitpunktwert) zwischen 0,0 und 1,0 normalisieren. Dann müssen Sie den Wert für den Mindest- und Höchstwert der darzustellenden physikalischen Einheit skalieren. Bei Werten in physikalischen Einheiten, die Sie in einen Analogausgangswert umwandeln müssen, normalisieren Sie zunächst den Wert in physikalischen Einheiten in einen Wert zwischen 0,0 und 1,0. Danach skalieren Sie den Wert zwischen 0 und 27648 oder -27648 und 27648, je nach Bereiche des Analogmoduls. STEP 7 bietet zu diesem Zweck die Anweisungen NORM\_X und SCALE\_X (Seite 231). Sie können auch die Anweisung CALCULATE (Seite 209) verwenden, um die Analogwerte (Seite 35) zu skalieren.

## 4.4 Datentypen

Datentypen geben die Größe eines Datenelements und die Art der Auswertung der Daten an. Jeder Anweisungsparameter unterstützt mindestens einen Datentyp, einige Parameter unterstützen mehrere Datentypen. Halten Sie den Mauszeiger auf dem Parameterfeld einer Anweisung, damit Ihnen angezeigt wird, welche Datentypen für den jeweiligen Parameter unterstützt werden.

Ein Formalparameter ist die Kennung an einer Anweisung, die die Adresse der von der Anweisung zu verwendenden Daten angibt (Beispiel: Eingang IN1 einer Anweisung ADD). Ein Aktualparameter ist die Adresse (mit vorangestelltem Zeichen "%") oder Konstante, die die von der Anweisung zu verwendenden Daten enthält (Beispiel: %MD400 "Anzahl\_Widgets"). Der Datentyp des von Ihnen angegebenen Aktualparameters muss einem der von der Anweisung angegebenen unterstützten Datentypen des Formalparameters entsprechen.

Wenn Sie einen Aktualparameter angeben, müssen Sie entweder eine Variable (Symbol) oder eine absolute (direkte) Adresse angeben. Variablen weisen einem symbolischen Namen (Variablennamen) einen Datentyp, Speicherbereich, Speicherversatz und Kommentar zu und können entweder im PLC-Variableneditor oder im Schnittstelleneditor eines Bausteins (OB, FC, FB und DB) angegeben werden. Wenn Sie eine absolute Adresse eingeben, die keine zugewiesene Variable hat, müssen Sie eine Größe verwenden, die einem unterstützten Datentyp entspricht, dann wird bei der Eingabe eine Standardvariable angelegt.

Alle Datentypen, mit Ausnahme des Datentyps String, sind im PLC-Variableneditor und in den Bausteinschnittstellen verfügbar. Der Datentyp String steht nur in den Bausteinschnittstellen zur Verfügung. Für viele der Eingangsparameter können Sie auch einen konstanten Wert eingeben.

- Bit und Bitfolgen (Seite 103): Bool (Boolescher Wert oder Bitwert), Byte (8-Bit-Bytewert), Word (16-Bit-Wortwert), DWord (32-Bit-Doppelwortwert)
- Ganzzahl (Seite 104)
  - USInt (vorzeichenlose 8-Bit-Ganzzahl), SInt (vorzeichenbehaftete 8-Bit-Ganzzahl),
  - UInt (vorzeichenlose 16-Bit-Ganzzahl), Int (vorzeichenbehaftete 16-Bit-Ganzzahl),
  - UDInt (vorzeichenlose 32-Bit-Ganzzahl), DInt (vorzeichenbehaftete 32-Bit-Ganzzahl),
- Gleitpunktzahl/Realzahl (Seite 104): Real (32-Bit-Real- oder Gleitpunktwert), LReal (64-Bit-Real- oder Gleitpunktwert)
- Uhrzeit und Datum (Seite 105): Time (32-Bit-IEC-Zeitwert), Date (16-Bit-Datumswert), TOD (32-Bit-Uhrzeitwert), DT (64-Bit-Datums-und-Zeitwert)
- Zeichen und Zeichenfolge (Seite 106): Char (8-Bit-Einzelzeichen), String (Zeichenfolge variabler Länge mit bis zu 254 Zeichen)
- Array (Seite 108)
- Datenstruktur (Seite 109): Struct
- PLC-Datentyp (Seite 109)
- Pointer (Seite 110): Pointer, Any, Variant

Ferner wird das folgende numerische BCD-Format von den Umwandlungsanweisungen unterstützt, obwohl es nicht als Datentyp zur Verfügung steht.

Tabelle 4- 15 Größe und Bereich des BCD-Formats

Format	Größe (Bit)	Bereich	Beispiel für konstanten Eintrag
BCD16	16	-999 bis 999	123, -123
BCD32	32	-9999999 bis 9999999	1234567, -1234567

#### 4.4.1 Datentypen Bool, Byte, Word und DWord

Tabelle 4- 16 Bit- und Bitfolge-Datentypen

Daten- typ	Bit- größe	Zahlen- typ	Zahlen- bereich	Beispiele für Konstanten	Beispiele für Adressen
Bool	1	Boolesch	FALSCH oder WAHR	WAHR, 1,	I1.0 Q0.1 M50.7 DB1.DBX2.3 Variablenname
		Binär	0 oder 1	0, 2#0	
		Oktal	8#0 oder 8#1	8#1	
		Hexadezimal	16#0 oder 16#1	16#1	
Byte	8	Binär	2#0 bis 2#11111111	2#00001111	IB2 MB10 DB1.DBB4 Variablenname
		Vorzeichenlose Ganzzahl	0 bis 255	15	
		Oktal	8#0 bis 8#377	8#17	
		Hexadezimal	B#16#0 bis B#16#FF	B#16#F, 16#F	
Word	16	Binär	2#0 bis 2#1111111111111111	2#1111000011110000	MW10 DB1.DBW2 Variablenname
		Vorzeichenlose Ganzzahl	0 bis 65535	61680	
		Oktal	8#0 bis 8#177777	8#170360	
		Hexadezimal	W#16#0 bis W#16#FFFF, 16#0 bis 16#FFFF	W#16#F0F0, 16#F0F0	
DWord	32	Binär	2#0 bis 2#11111111111111111111111111111111	2#1111000011111111100 001111	MD10 DB1.DBD8 Variablenname
		Vorzeichenlose Ganzzahl	0 bis 4294967295	15793935	
		Oktal	8#0 bis 8#3777777777	8#74177417	
		Hexadezimal	DW#16#0000_0000 bis DW#16#FFFF_FFFF, 16#0000_0000 bis 16#FFFF_FFFF	DW#16#F0FF0F, 16#F0FF0F	

### 4.4.2 Ganzzahlige Datentypen

Tabelle 4- 17 Ganzzahlige Datentypen (U = vorzeichenlos, S = kurz, D = doppelt)

Datentyp	Bitgröße	Zahlenbereich	Beispiele für Konstanten	Beispiele für Adressen
USInt	8	0 bis 255	78, 2#01001110	MB0, DB1.DBB4, Variablenname
SInt	8	-128 bis 127	+50, 16#50	
UInt	16	0 bis 65.535	65295, 0	MW2, DB1.DBW2, Variablenname
Int	16	-32.768 bis 32.767	30000, +30000	
UDInt	32	0 bis 4.294.967.295	4042322160	MD6, DB1.DBD8, Variablenname
DInt	32	-2.147.483.648 bis 2.147.483.647	-2131754992	

### 4.4.3 Gleitpunktzahl/Realzahl-Datentypen

Realzahlen (bzw. Gleitpunktzahlen) werden als einfachgenaue 32-Bit-Zahlen (Real) oder als doppelgenaue 64-Bit-Zahlen (LReal) dargestellt, wie in der Norm ANSI/IEEE7541985 beschrieben. Einfachgenaue Gleitpunktzahlen sind bis zu 6 signifikante Ziffern genau und doppelgenaue Gleitpunktzahlen sind bis zu 15 signifikante Ziffern genau. Sie können maximal 6 (Real) oder 15 (LReal) signifikante Ziffern angeben, wenn Sie eine Gleitpunktkonstante eingeben.

Tabelle 4- 18 Gleitpunktzahl/Realzahl-Datentypen (L = lang)

Datentyp	Bitgröße	Zahlenbereich	Beispiele für Konstanten	Beispiele für Adressen
Real	32	-3.402823e+38 bis -1.175 495e-38, ±0, +1,175 495e-38 bis +3,402823e+38	123.456, -3.4, 1,0e-5	MD100, DB1.DBD8, Variablenname
LReal	64	-1,7976931348623158e+308 bis -2,2250738585072014e-308, ±0, +2,2250738585072014e-308 bis +1,7976931348623158e+308	12345,123456789e40, 1,2E+40	DB-Name.Var-Name Regeln: <ul style="list-style-type: none"> <li>• Direkte Adressierung wird nicht unterstützt</li> <li>• Kann in der Bausteinschnittstelle eines OBs, FBs oder FCs zugewiesen werden</li> </ul>

Bei Berechnungen, die eine lange Reihe von Werten einschließlich sehr großen und sehr kleinen Zahlen benötigen, kann es zu ungenauen Ergebnissen kommen. Dies kann auftreten, wenn sich die Zahlen um  $10^x$  unterscheiden, wobei  $x > 6$  (Real) oder 15 (LReal) ist. Beispiel (Real):  $100\ 000\ 000 + 1 = 100\ 000\ 000$ .



#### 4.4.4 Uhrzeit- und Datums-Datentypen

Tabelle 4- 19 Uhrzeit- und Datums-Datentypen

Datentyp	Größe	Bereich	Beispiele für konstanten Eintrag
Time	32 Bits	T#-24d_20h_31m_23s_648ms bis T#24d_20h_31m_23s_647ms Gespeichert als: -2.147.483.648 ms bis +2.147.483.647 ms	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
Date	16 Bits	D#1990-1-1 bis D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
Time_of_Day	32 Bit	TOD#0:0:0.0 bis TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (Date and Time Long)	12 Byte	Min.: DTL#1970-01-01-00:00:00.0 Max.: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

#### Zeit

TIME-Daten werden als vorzeichenbehaftete doppelte Ganzzahl gespeichert, die als Millisekunden ausgewertet wird. Das Editorformat kann Angaben für Tag (d), Stunden (h), Minuten (m), Sekunden (s) und Millisekunden (ms) verarbeiten.

Es ist nicht erforderlich, alle Zeiteinheiten anzugeben. So sind z. B. T#5h10s und 500h gültig.

Der Gesamtwert aller angegebenen Werte in den einzelnen Einheiten darf den oberen bzw. unteren Grenzwert in Millisekunden für den Zeit-Datentyp nicht über- bzw. unterschreiten (- 2.147.483.648 ms bis +2.147.483.647 ms).

#### Datum

DATE-Daten werden als vorzeichenloser Ganzzahlenwert gespeichert, der als Anzahl der Tage ausgewertet wird, die zum Basisdatum 01.01.1990 addiert werden, um das angegebene Datum zu erhalten. Das Editorformat muss ein Jahr, einen Monat und einen Tag angeben.

#### TOD

TOD-Daten (TIME\_OF\_DAY) werden als vorzeichenlose doppelte Ganzzahl gespeichert, die als Anzahl der Millisekunden seit Mitternacht für die angegebene Uhrzeit ausgewertet wird (Mitternacht = 0 ms). Es müssen die Stunden (24 h/Tag), Minuten und Sekunden angegeben werden. Die Angabe der Nachkommastellen der Sekunde ist optional.

**DTL**

Der DTL-Datentyp (Date and Time Long) nutzt eine 12-Byte-Struktur, um Angaben zum Datum und zur Uhrzeit zu speichern. Sie können den Datentyp DTL entweder im temporären Speicher eines Bausteins oder in einem DB definieren. Für alle Komponenten muss in der Spalte für den Startwert im DB-Editor ein Wert eingegeben werden.

Tabelle 4- 20 Größe und Bereich von DTL

Länge (Bytes)	Format	Wertebereich	Beispiel für Werteingabe
12	Uhrzeit und Kalender Jahr-Monat-Tag:Stunde:Minute: Sekunde.Nanosekunden	Min.: DTL#1970-01-01-00:00:00.0 Max.: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

Jede Komponente des Datentyps DTL enthält einen unterschiedlichen Datentyp und Wertebereich. Der Datentyp eines angegebenen Werts muss dem Datentyp der jeweiligen Komponenten entsprechen.

Tabelle 4- 21 Elemente der DTL-Struktur

Byte	Komponente	Datentyp	Wertebereich
0	Jahr	UINT	1970 bis 2554
1			
2	Monat	USINT	1 bis 12
3	Tag	USINT	1 bis 31
4	Wochentag <sup>1</sup>	USINT	1 (Sonntag) bis 7 (Samstag) <sup>1</sup>
5	Stunde	USINT	0 bis 23
6	Minute	USINT	0 bis 59
7	Sekunde	USINT	0 bis 59
8	Nanosekunden	UDINT	0 bis 999.999.999
9			
10			
11			

<sup>1</sup> Der Wochentag wird im Werteintrag nicht berücksichtigt.

**4.4.5 Zeichen- und Zeichenfolge-Datentypen**

Tabelle 4- 22 Zeichen- und Zeichenfolge-Datentypen

Datentyp	Größe	Bereich	Beispiel für konstanten Eintrag
Char	8 Bit	ASCII-Zeichencodes: 16#00 bis 16#FF	'A', 't', '@'
String	n+ 2 Byte	n = (0 bis 254 Zeichenbyte)	'ABC'

## Char

Char-Daten belegen ein Byte im Speicher und speichern ein einzelnes Zeichen, das im ASCII-Format codiert ist. In der Editorsyntax wird ein einzelnes Hochkomma vor und hinter dem ASCII-Zeichen eingegeben. Sichtbare Zeichen und Steuerzeichen sind verwendbar. Eine Tabelle mit den gültigen Steuerzeichen finden Sie in der Beschreibung des Datentyps Zeichen.

## String

Die CPU unterstützt den Datentyp String zum Speichern einer Folge von Einzelbyte-Zeichen. Der Datentyp String enthält die Gesamtzeichenzahl (Anzahl der Zeichen in der Zeichenkette) und die tatsächliche Zeichenzahl. Der Datentyp String bietet bis zu 256 Bytes zum Speichern der maximalen Gesamtzeichenzahl (1 Byte), der tatsächlichen Zeichenzahl (1 Byte) und bis zu 254 Zeichen, wobei jedes Zeichen in 1 Byte abgelegt wird.

Sie können literale Zeichenketten (Konstanten) für Anweisungsparameter vom Typ IN in einzelnen Hochkommata angeben. 'ABC' zum Beispiel ist eine Zeichenkette aus drei Zeichen, die als Eingang für Parameter IN der Anweisung S\_CONV genutzt werden kann. Sie können auch Zeichenkettenvariablen erstellen, indem Sie in der Bausteinschnittstelle von OB, FC, FB und DB den Datentyp "String" auswählen. Im PLC-Variableneditor können Sie keine Zeichenkette erstellen.

Sie können die maximale Zeichenfolgegröße in Byte mithilfe von eckigen Klammern nach dem Schlüsselwort "String" angeben (sobald Sie den Datentyp "String" in einer Datentyp-Klappliste ausgewählt haben). Beispiel: "MeinString String[10]" gibt eine maximale Größe von 10 Byte für die Zeichenfolge MeinString an. Wenn Sie die eckigen Klammern mit der Angabe der maximalen Größe weglassen, wird die Größe 254 angenommen.

Das folgende Beispiel zeigt eine Zeichenfolge mit der maximalen Zeichenzahl 10 und der tatsächlichen Zeichenzahl 3. Die Zeichenfolge enthält somit 3 Zeichen mit je einem Byte, kann aber bis auf 10 Zeichen mit je einem Byte erweitert werden.

Tabelle 4- 23 Beispiel für einen Zeichenfolge-Datentyp

Gesamtzeichenzahl	Tatsächliche Zeichenzahl	Zeichen 1	Zeichen 2	Zeichen 3	...	Zeichen 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	...	Byte 11

ASCII-Steuerzeichen können in Char- und String-Daten verwendet werden. Die folgende Tabelle zeigt Beispiele für die Syntax von Steuerzeichen.

Tabelle 4- 24 Gültige ASCII-Steuerzeichen

Steuerzeichen	ASCII-Hexadezimalwert	Regelungsfunktion	Beispiele
\$L oder \$l	0A	Zeilenvorschub	'\$LText', '\$0AText'
\$N oder \$n	0A und 0D	Zeilenumbuch Die neue Zeile zeigt zwei Zeichen in der Zeichenfolge.	'\$NText', '\$0A\$0DText'
\$P oder \$p	0C	Formularvorschub	'\$PText', '\$0CText'

Steuerzeichen	ASCII-Hexadezimalwert	Regelungsfunktion	Beispiele
\$R oder \$r	0D	Zeilenschaltung (CR)	'\$RText', '\$0DText'
\$T oder \$t	09	Tabulator	'\$TText', '\$09Text'
\$\$	24	Dollarzeichen	'100\$\$', '100\$24'
\$'	27	Einzelnes Hochkomma	'\$'Text\$', '\$27Text\$27'

### 4.4.6 Datentyp ARRAY

#### Arrays

Sie können ein Array erstellen, das mehrere Elemente des gleichen Datentyps enthält. Arrays können in der Bausteinschnittstelle von OB, FC, FB und DB angelegt werden. Im PLC-Variableneditor können Sie kein Array erstellen.

Um ein Array in der Bausteinschnittstelle zu erstellen, benennen Sie das Array und wählen den Datentyp "Array [lo .. hi] of type", dann ändern Sie "lo", "hi" und "type" wie folgt:

- lo - Anfangsindex (niedrigster Index) für Ihr Array
- hi - Abschlussindex (höchster Index) für Ihr Array
- type - einer der Datentypen wie BOOL, SINT, UDINT

Tabelle 4- 25 Regeln für den Datentyp ARRAY

Datentyp	Array-Syntax		
ARRAY	Name [index1_min..index1_max, index2_min..index2_max] of <Datentyp>		
	<ul style="list-style-type: none"> <li>• Alle Array-Elemente müssen den gleichen Datentyp haben.</li> <li>• Der Index kann negativ sein, doch der untere Grenzwert muss kleiner oder gleich dem oberen Grenzwert sein.</li> <li>• Arrays können bis zu sechs Dimensionen umfassen.</li> <li>• Multidimensionale Indexdeklarationen min..max werden durch Kommazeichen voneinander getrennt.</li> <li>• Verschachtelte Arrays bzw. Arrays aus Arrays sind nicht zulässig.</li> <li>• Die Speichergröße eines Arrays = (Größe eines Elements * Gesamtzahl der Elemente im Array)</li> </ul>		
	Arrayindex	Gültige Index-Datentypen	Regeln für den Arrayindex
Konstante oder Variable	USInt, SInt, UInt, Int, UDInt, DInt	<ul style="list-style-type: none"> <li>• Grenzwerte: -32768 bis +32767</li> <li>• Gültig: Gemischte Konstanten und Variablen</li> <li>• Gültig: Konstante Ausdrücke</li> <li>• Ungültig: Variable Ausdrücke</li> </ul>	

<b>Beispiel: Array-Deklarationen</b>	ARRAY[1..20] of REAL	Eine Dimension, 20 Elemente
	ARRAY[-5..5] of INT	Eine Dimension, 11 Elemente
	ARRAY[1..2, 3..4] of CHAR	Zwei Dimensionen, 4 Elemente
<b>Beispiel: Array-Adressen</b>	ARRAY1[0]	ARRAY1 Element 0
	ARRAY2[1,2]	ARRAY2 Element [1,2]
	ARRAY3[i,j]	Wenn i = 3 und j = 4, dann wird ARRAY3 Element [3, 4] angesprochen

#### 4.4.7 Datentyp Struktur

Mit dem Datentyp "Struct" können Sie eine aus anderen Datentypen bestehende Datenstruktur definieren. Der Datentyp Struct kann genutzt werden, um eine Gruppe zusammengehöriger Prozessdaten als eine Dateneinheit zu behandeln. Der Datentyp Struct wird benannt und die interne Datenstruktur im Datenbausteineditor oder in einem Bausteinschnittstelleneditor deklariert.

Arrays und Strukturen können auch zu einer größeren Struktur zusammengefügt werden. Eine Struktur kann bis zu acht Ebenen tief verschachtelt werden. Sie können z. B. eine Struktur aus Strukturen erstellen, die wiederum Arrays enthalten.

Eine Struct-Variable beginnt an der Adresse eines geraden Bytes und belegt den Speicher bis zur nächsten Wortgrenze.

#### 4.4.8 PLC-Datentyp

Im PLC-Datentypeditor können Sie Datenstrukturen definieren, die Sie mehrmals in Ihrem Programm verwenden können. Sie erstellen einen PLC-Datentyp durch Öffnen von "PLC-Datentypen" in der Projektnavigation und Doppelklick auf den Befehl "Neuen Datentyp hinzufügen". Klicken Sie zweimal einzeln auf den neu erstellen PLC-Datentyp, um den Standardnamen zu ändern. Doppelklicken Sie dann, um den PLC-Datentypeditor zu öffnen.

Zum Erstellen einer benutzerdefinierten PLC-Datentypstruktur können Sie auf dieselbe Weise vorgehen wie im Datenbausteineditor. Fügen Sie für alle zusätzlich erforderlichen Datentypen neue Zeilen ein, um die gewünschte Datenstruktur anzulegen.

Wenn Sie einen PLC-Datentyp anlegen, erscheint der Name des neuen PLC-Datentyp in der Auswahl-Klappliste im DB-Editor und im Codebaustein-Schnittstelleneditor.

Mögliche Verwendung von PLC-Datentypen:

- PLC-Datentypen können direkt als Datentyp in einer Codebausteinschnittstelle oder in Datenbausteinen verwendet werden.
- PLC-Datentypen können als Vorlage für die Erstellung von mehreren globalen Datenbausteinen mit der gleichen Datenstruktur verwendet werden.

Ein PLC-Datentyp kann beispielsweise ein Rezept zum Mischen von Farben sein. Sie können diesen PLC-Datentyp dann mehreren Datenbausteinen zuweisen. In jedem Datenbaustein können die Variablen dann entsprechend angepasst werden, um eine bestimmte Farbe zu erstellen.

### 4.4.9 Pointer-Datentypen

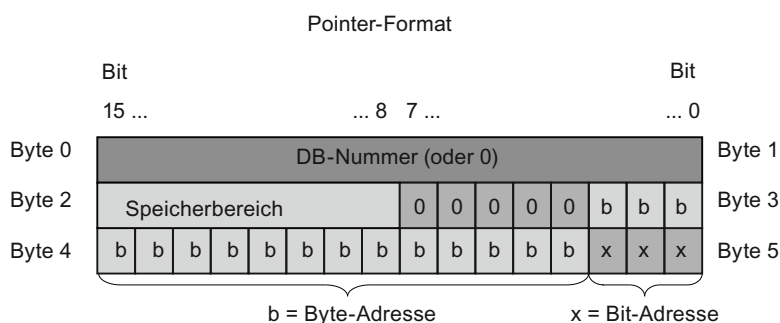
Die Pointer-Datentypen (Pointer, Any und Variant) können in den Bausteinschnittstellentabellen für FB- und FC-Codebausteine verwendet werden. Sie können in der Datentyp-Klappliste der Bausteinschnittstelle einen Pointer-Datentyp auswählen.

Der Datentyp Variant wird auch für Anweisungsparameter verwendet.

#### 4.4.9.1 Pointer-Datentyp "Pointer"

Der Datentyp Pointer zeigt auf eine bestimmte Variable. Der Datentyp belegt 6 Bytes (48 Bits) im Speicher und kann die folgenden Informationen enthalten:

- DB-Nummer oder 0, wenn die Daten nicht in einem DB gespeichert werden
- Speicherbereich in der CPU
- Variablenadresse



Je nach Anweisung können Sie die folgenden drei Typen von Pointern deklarieren:

- Bereichsinterner Pointer: enthält Daten zur Adresse einer Variable
- Bereichsübergreifender Pointer: enthält Daten zum Speicherbereich und zur Adresse einer Variable
- DB-Pointer: enthält eine Datenbausteinnummer und die Adresse einer Variable

Tabelle 4- 26 Pointer-Typen:

Typ	Format	Beispieleintrag
Bereichsinterner Pointer	P#Byte.Bit	P#20.0
Bereichsübergreifender Pointer	P#Speicherbereich_Byte.Bit	P#M20.0
DB-Pointer	P#Datenbaustein.Datenelement	P#DB10.DBX20.0

Einen Parameter vom Typ Pointer können Sie ohne Präfix (P #) eingeben. Ihre Eingabe wird automatisch ins Pointer-Format umgewandelt.

Tabelle 4- 27 Codierung des Speicherbereichs in den Pointer-Daten:

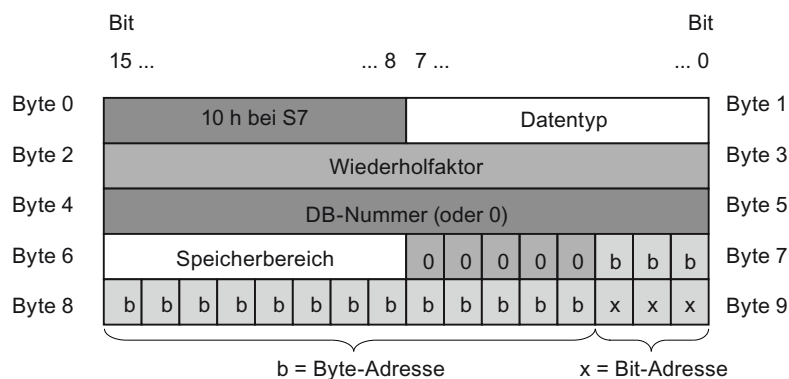
Hexadezimalcode	Datentyp	Beschreibung
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DBX	Datenbaustein
b#16#85	DIX	Instanz-Datenbaustein
b#16#86	L	Lokaldaten
b#16#87	V	Vorherige Lokaldaten

#### 4.4.9.2 Pointer-Datentyp "Any"

Der Pointer-Datentyp ANY ("Any") zeigt auf den Anfang eines Datenbereichs und gibt dessen Länge an. Der Pointer ANY belegt 10 Bytes im Speicher und kann die folgenden Informationen enthalten:

- Datentyp: Datentyp der Datenelemente
- Wiederholfaktor: Anzahl der Datenelemente
- DB-Nummer: Datenbaustein, in dem Datenelemente gespeichert werden
- Speicherbereich: Speicherbereich der CPU, in der die Datenelemente gespeichert werden
- Anfangsadresse: "Byte.Bit"-Anfangsadresse der Daten

Die folgende Abbildung zeigt die Struktur des Pointes ANY:



Ein Pointer kann keine ANY-Strukturen erkennen. Er kann nur lokalen Variablen zugewiesen werden.

Tabelle 4- 28 Format und Beispiele des Pointers ANY:

Format	Beispieleintrag	Beschreibung
P#Datenbaustein.Speicherbereich Datenadresse Typ Nummer	P#DB 11.DBX 20.0 INT 10	10 Wörter im globalen DB 11 mit Beginn an DBB 20.0
P#Speicherbereich Datenadresse Typ Nummer	P#M 20.0 BYTE 10	10 Byte mit Beginn an MB 20.0
	P#E 1.0 BOOL 1	Eingang E1.0

Tabelle 4- 29 Codierung des Datentyps im Pointer ANY

Hexadezimalcode	Datentyp	Beschreibung
b#16#00	Null	Null-Pointer
b#16#01	Bool	Bits
b#16#02	Byte	Bytes, 8 Bits
b#16#03	Char	8-Bit-Zeichen
b#16#04	Word	16-Bit-Wort
b#16#05	Int	16-Bit-Ganzzahl
b#16#37	SInt	8-Bit-Ganzzahl
b#16#35	UInt	vorzeichenlose 16-Bit-Ganzzahl
b#16#34	USInt	vorzeichenlose 8-Bit-Ganzzahl
b#16#06	DWord	32-Bit-Doppelwort
b#16#07	DInt	32-Bit-Ganzzahl
b#16#36	UDInt	vorzeichenlose 32-Bit-Ganzzahl
b#16#08	Real	32-Bit-Gleitpunkt
b#16#0B	Time	Zeit
b#16#13	String	Zeichenfolge

Tabelle 4- 30 Codierung des Speicherbereichs im Pointer ANY:

Hexadezimalcode	Speicherbereich	Beschreibung
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DBX	Datenbaustein
b#16#85	DIX	Instanz-Datenbaustein
b#16#86	L	Lokaldaten
b#16#87	V	Vorherige Lokaldaten



### 4.4.9.3 Pointer-Datentyp "Variant"

Der Datentyp Variant kann auf Variablen verschiedener Datentypen oder Parameter zeigen. Der Pointer Variant kann auf Strukturen und einzelne Strukturkomponenten zeigen. Der Pointer Variant belegt keinen Platz im Speicher.

Tabelle 4- 31 Eigenschaften des Pointers Variant

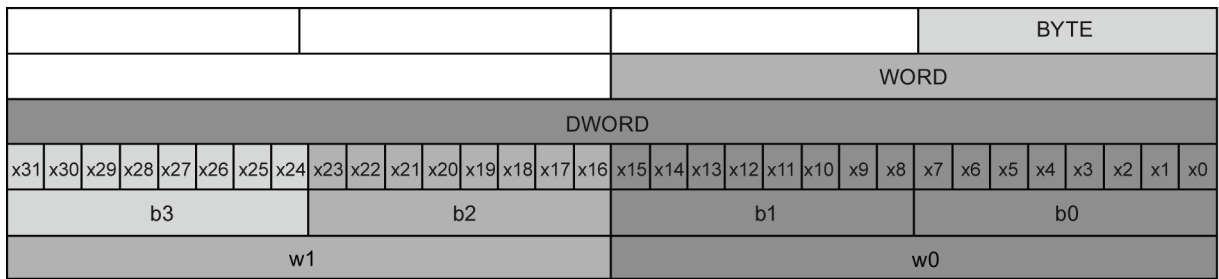
Länge (Byte)	Darstellung	Format	Beispieleintrag
0	Symbolischer	Operand	MeineVariable
		DB-Name.Strukturname.Elementname	MeinDB.Strukt1.Druck1
	Absoluter	Operand	%MW10
		DB-Nummer.Operand Typ Länge	P#DB10.DBX10.0 INT 12

### 4.4.10 Zugriff auf eine "Scheibe" eines Variablendatentyps

Auf PLC-Variablen und Datenbausteinvariablen kann je nach ihrer Größe auf Bit-, Byte- oder Wortebene zugegriffen werden. Die Syntax für den Zugriff auf eine Datenscheibe lautet wie folgt:

- "<PLC-Variablenname>".xn (Bitzugriff)
- "<PLC-Variablenname>".bn (Bytezugriff)
- "<PLC-Variablenname>".wn (Wortzugriff)
- "<Datenbausteinname>".<Variablenname>.xn (Bitzugriff)
- "<Datenbausteinname>".<Variablenname>.bn (Bytezugriff)
- "<Datenbausteinname>".<Variablenname>.wn (Wortzugriff)

Auf eine Variable mit Doppelwortgröße kann über die Bits 0-31, Bytes 0-3 oder die Wörter 0-1 zugegriffen werden. Auf eine Variable mit Wortgröße kann über die Bits 0-15, Bytes 0-2 oder das Wort 0 zugegriffen werden. Auf eine Variable mit Bytegröße kann über die Bits 0-8 oder das Byte 0 zugegriffen werden. Bit-, Byte- und Wort-Scheiben können überall dort verwendet werden, wo Bits, Bytes oder Wörter erwartete Operanden sind.



**Hinweis**

Auf folgende Datentypen kann über Scheiben zugegriffen werden: Byte, Char, Conn\_Any, Date, DInt, DWord, Event\_Any, Event\_Att, Hw\_Any, Hw\_Device, HW\_Interface, Hw\_Io, Hw\_Pwm, Hw\_SubModule, Int, OB\_Any, OB\_Att, OB\_Cyclic, OB\_Delay, OB\_WHINT, OB\_PCYCLE, OB\_STARTUP, OB\_TIMEERROR, OB\_Tod, Port, Rtm, SInt, Time, Time\_Of\_Day, UDIInt, UInt, USInt und Word. Auf PLC-Variablen vom Datentyp Real kann über Scheiben zugegriffen werden, auf Datenbausteinvariablen vom Typ Real jedoch nicht.

**Beispiele**

In der PLC-Variablen-tabelle ist "DW" eine deklarierte Variable vom Typ DWORD. Die Beispiele zeigen den Zugriff in Form von Bit-, Byte- und Wortscheiben:

	KOP	FUP	SCL
<b>Bitzugriff</b>	<p>"DW".x11</p>		<pre>IF "DW".x11 THEN ... END_IF;</pre>
<b>Bytezugriff</b>	<p>"DW".b2</p> <p>==</p> <p>Byte</p> <p>"DW".b3</p>		<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
<b>Wortzugriff</b>			<pre>out:= "DW".w0 AND "DW".w1;</pre>

**Siehe auch**

SCL (Seite 166)

#### 4.4.11 Zugriff auf eine Variable mit einer AT-Überlagerung

Mit Hilfe der AT-Variablenüberlagerung können Sie mit einer überlagerten Deklaration eines unterschiedlichen Datentyps auf eine bereits deklarierte Variable eines Standardzugriffsbausteins zugreifen. Sie können beispielsweise die einzelnen Bits einer Variable vom Datentyp Byte, Word oder DWord mit einem Bool-Array adressieren.

##### Deklaration

Um einen Parameter zu überlagern, deklarieren Sie einen zusätzlichen Parameter direkt nach dem zu überlagernden Parameter und wählen den Datentyp "AT". Der Editor legt die Überlagerung an und Sie können dann den Datentyp, die Struktur oder das Array für die Überlagerung wählen.

##### Beispiel

Dieses Beispiel zeigt die Eingangsparameter eines FBs mit Standardzugriff. Die Bytevariable B1 wird mit einem Booleschen Array überlagert:

■	B1	Byte
▼	AT	AT "B1" Array [0..7] of Bool
■	AT[0]	Bool
■	AT[1]	Bool
■	AT[2]	Bool
■	AT[3]	Bool
■	AT[4]	Bool
■	AT[5]	Bool
■	AT[6]	Bool
■	AT[7]	Bool



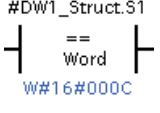



Tabelle 4- 32 Überlagerung eines Bytes mit einem Booleschen Array

7	6	5	4	3	2	1	0
AT[0]	AT[1]	AT[2]	AT[3]	AT[4]	AT[5]	AT[6]	AT[7]

Ein weiteres Beispiel ist eine Variable vom Typ DWord, die mit einer Struktur überlagert wird:

■	DW1	DWord
▼	DW1_Struct	AT "DW1" Struct
■	S1	Word
■	S2	Byte
■	S3	Byte

Die Überlagerungstypen können in der Programmlogik direkt angesprochen werden:

KOP	FUP	SCL
		<pre>IF #AT[1] THEN ... END_IF;</pre>
		<pre>IF (#DW1_Struct.S1 = W#16#000C) THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.S2;</pre>

**Regeln**

- Die Überlagerung von Variablen ist nur in FB- und FC-Bausteinen mit Standardzugriff möglich.
- Sie können Parameter für alle Baustein Typen und alle Deklarationsabschnitte überlagern.
- Ein überlagerter Parameter kann wie jeder andere Bausteinparameter verwendet werden.
- Parameter vom Typ VARIANT können Sie nicht überlagern.
- Die Größe des überlagernden Parameters muss kleiner oder gleich der Größe des überlagerten Parameters sein.
- Die überlagernde Variable muss sofort nach der Variablen deklariert werden, die sie überlagert, und muss durch das Schlüsselwort "AT" gekennzeichnet sein.

**Siehe auch**

SCL (Seite 166)

**4.5 Memory Card verwenden**

<b>ACHTUNG</b>
Die CPU unterstützt nur die vorformatierte SIMATIC Memory Card (Seite 875). Löschen Sie vor dem Kopieren von Programmen auf die formatierte Memory Card alle zuvor gespeicherten Programme von der Memory Card.

Sie können die Memory Card als Übertragungskarte oder als Programmkarte nutzen. Jedes Programm, das Sie auf die Memory Card kopieren, enthält alle Codebausteine und Datenbausteine, alle Technologieobjekte und die Gerätekonfiguration. Ein kopiertes Programm enthält **keine** geforcten Werte.

- Mit einer Übertragungskarte (Seite 119) kopieren Sie ein Programm in den internen Ladespeicher der CPU, ohne dafür STEP 7 zu verwenden. Nachdem Sie die Übertragungskarte gesteckt haben, löscht die CPU zunächst das Anwenderprogramm und alle geforcten Werte aus dem internen Ladespeicher und kopiert dann das Programm von der Übertragungskarte in den internen Ladespeicher. Wenn der Übertragungsvorgang beendet ist, müssen Sie die Übertragungskarte ziehen.

Mit einer leeren Übertragungskarte können Sie auf eine passwortgeschützte CPU zugreifen, wenn Sie das Passwort verloren oder vergessen haben (Seite 127). Durch Stecken der leeren Übertragungskarte wird das passwortgeschützte Programm im internen Ladespeicher der CPU gelöscht. Dann können Sie ein neues Programm in die CPU laden.

- Eine Programmkarte (Seite 121) nutzen Sie als externen Ladespeicher für die CPU. Wenn Sie eine Programmkarte in die CPU stecken, wird der gesamte interne Ladespeicher der CPU gelöscht (Anwenderprogramm und ggf. geforcten Werte). Dann führt die CPU das Programm im externen Ladespeicher aus (Programmkarte). Beim Laden in eine CPU mit Programmkarte wird nur der externe Ladespeicher (die Programmkarte) aktualisiert.

Weil der interne Ladespeicher der CPU beim Stecken der Programmkarte gelöscht wurde, **muss** die Programmkarte in der CPU gesteckt bleiben. Wenn Sie die Programmkarte ziehen, geht die CPU in den Betriebszustand STOP. (Die Fehler-LED blinkt, um anzuzeigen, dass die Programmkarte entfernt wurde.)

Das kopierte Programm auf einer Memory Card umfasst die Codebausteine, Datenbausteine, Technologieobjekte und die Gerätekonfiguration. Die Memory Card enthält **keine** geforcten Werte. Die geforcten Werte sind nicht Teil des Programms, doch sie werden im Ladespeicher abgelegt. Dies kann der interne Ladespeicher der CPU oder der externe Ladespeicher (eine Programmkarte) sein. Wenn eine Programmkarte in der CPU steckt, wendet STEP 7 die geforcten Werte nur auf den externen Ladespeicher auf der Programmkarte an.

Sie verwenden die Memory Card auch dann, wenn Sie Firmware-Updates (Seite 124) herunterladen.

### 4.5.1 Memory Card in die CPU stecken

#### VORSICHT

Elektrostatistische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.



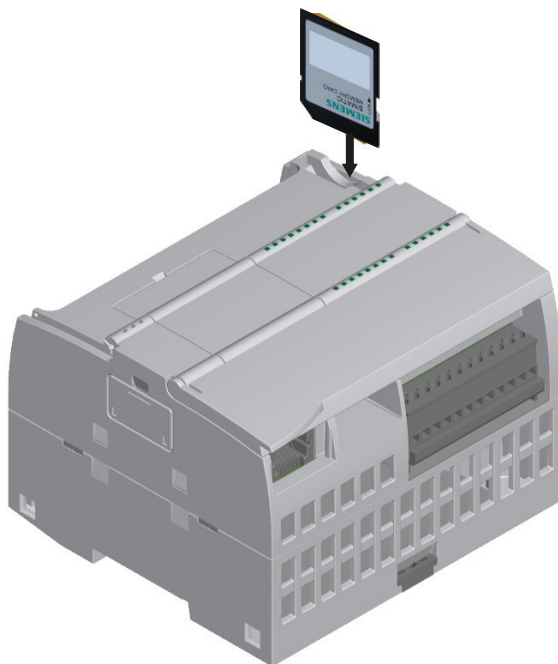
Stellen Sie sicher, dass die Memory Card nicht schreibgeschützt ist. Schieben Sie dazu den Schutzschalter aus der Verriegelungsposition heraus.

#### VORSICHT

Wenn Sie eine Memory Card (unabhängig davon, ob als Programm- oder als Übertragungskarte genutzt) in eine laufende CPU stecken, geht die CPU sofort in den Betriebszustand STOP, was zu Sachschaden an den gesteuerten Geräten oder im gesteuerten Prozess führen kann. Stellen Sie vor dem Stecken oder Ziehen einer Memory Card stets sicher, dass die CPU nicht aktiv eine Maschine oder einen Prozess steuert. Installieren Sie einen NOT-AUS-Schaltkreis für Ihre Anwendung bzw. Ihren Prozess.

#### Hinweis

Wenn Sie eine Memory Card im Betriebszustand STOP der CPU stecken, zeigt der Diagnosepuffer die Meldung an, dass die Auswertung der Memory Card gestartet wurde. Die CPU wertet die Memory Card aus, wenn Sie entweder die CPU in den Betriebszustand RUN versetzen, den Speicher der CPU über MRES urlöschen oder die CPU aus- und wieder einschalten.

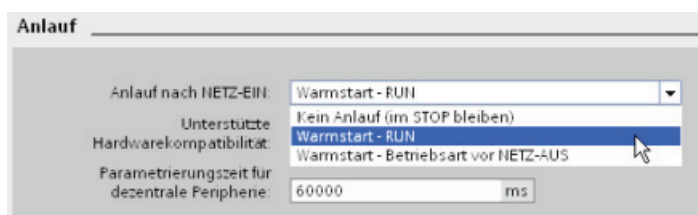


Zum Einsetzen einer Memory Card öffnen Sie die obere Abdeckung der CPU und stecken die Memory Card in den Steckplatz. Ein Steckverbinder ermöglicht einfaches Stecken und Ziehen des Moduls.

Die Memory Card ist so geformt, dass sie nur in eine Richtung in den Schacht gesteckt werden kann.

## 4.5.2 Anlaufparameter der CPU vor dem Kopieren des Projekts auf die Memory Card konfigurieren

Wenn Sie ein Programm auf eine Übertragungskarte oder eine Programmkarte kopieren, enthält das Programm die Anlaufparameter für die CPU. Stellen Sie stets vor dem Kopieren des Programms auf die Memory Card sicher, dass Sie den Betriebszustand der CPU nach dem Aus- und wieder Einschalten konfiguriert haben. Sie können wählen, ob die CPU im Betriebszustand STOP oder RUN oder im letzten Betriebszustand (vor dem Neustart) starten soll.



## 4.5.3 Einsatz der Memory Card als Übertragungskarte

### VORSICHT

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.

## Übertragungskarte anlegen

Denken Sie daran, die Anlaufparameter der CPU zu konfigurieren (Seite 119), bevor Sie ein Programm auf die Übertragungskarte kopieren. Um eine Übertragungskarte anzulegen, gehen Sie folgendermaßen vor:

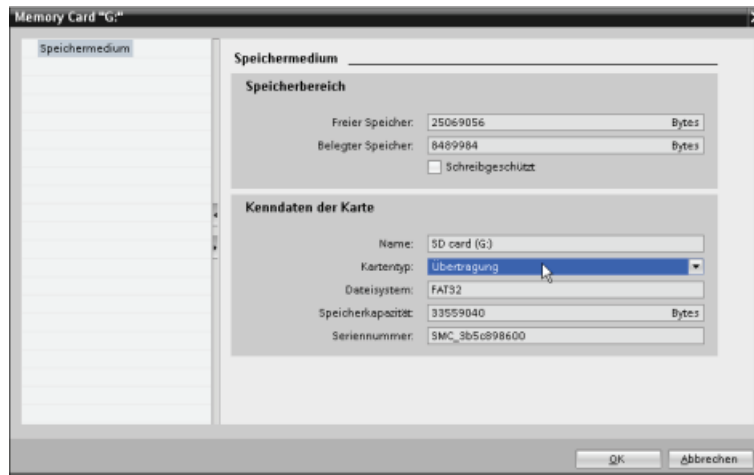
1. Stecken Sie eine leere SIMATIC Memory Card in einen SD-Kartenleser, der an Ihren Computer angeschlossen ist.

Wenn Sie eine SIMATIC Memory Card verwenden, die bereits ein Anwenderprogramm oder ein Firmware-Update enthält, **müssen** Sie die Programmdateien löschen, bevor Sie die Karte erneut verwenden. Rufen Sie den Windows Explorer auf und zeigen Sie die Inhalte der Memory Card an. Löschen Sie die Datei "S7\_JOB.S7S" sowie alle vorhandenen Ordner mit Datenprotokollen und alle Verzeichnisordner (wie "SIMATIC.S7S" oder "FWUPDATE.S7S").

2. Erweitern Sie in der Projektnavigation (Projektansicht) den Ordner "SIMATIC Kartenleser" und wählen Sie Ihren Kartenleser aus.

3. Klicken Sie mit der rechten Maustaste auf den Laufwerksbuchstaben der Memory Card im Kartenleser und wählen Sie im Kontextmenü die Option "Eigenschaften". Daraufhin wird der Dialog "Memory Card" angezeigt.
4. Wählen Sie im Dialog "Memory Card" in der Klappliste "Kartentyp" die Option "Übertragen" aus.

Daraufhin erstellt STEP 7 die leere Übertragungskarte. Wenn Sie eine leere Übertragungskarte anlegen, weil Sie Ihr CPU-Passwort verloren haben (Seite 127), entnehmen Sie die Übertragungskarte aus dem Kartenleser.



5. Fügen Sie das Programm hinzu, indem Sie in der Projektnavigation die CPU (z.B. PLC\_1 [CPU 1214 DC/DC/DC]) auswählen und mit der Maus auf die Memory Card ziehen. (Alternativ können Sie die CPU kopieren und in die Memory Card einfügen.) Durch Kopieren der CPU in die Memory Card wird der Dialog "Vorschau laden" geöffnet.
6. Klicken Sie im Dialog "Vorschau laden" auf die Schaltfläche "Laden", um die CPU in die Memory Card zu kopieren.
7. Wenn der Dialog eine Meldung anzeigt, dass die CPU (das Programm) fehlerfrei geladen wurde, klicken Sie auf die Schaltfläche "Fertig stellen".

## Übertragungskarte verwenden

### WARNUNG

**Vergewissern Sie sich vor dem Einlegen der Memory Card, dass die CPU keinen Prozess ausführt.**

Durch das Einlegen einer Memory Card geht die CPU in den Betriebszustand STOP, was sich auf den Betrieb eines Online-Prozesses oder einer Maschine auswirken kann. Unvorhersehbarer Betrieb eines Prozesses oder einer Maschine kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Stellen Sie vor dem Einlegen einer Memory Card sicher, dass die CPU offline und in einem sicheren Zustand ist.



Um das Programm in die CPU zu übertragen, gehen Sie folgendermaßen vor:

1. Stecken Sie die Übertragungskarte in die CPU (Seite 117). Wenn sich die CPU in RUN befindet, geht die CPU in den Betriebszustand STOP. Die Wartungs-LED (MAINT) blinkt, um anzuzeigen, dass die Memory Card ausgewertet werden muss.
2. Schalten Sie die CPU aus und wieder ein, um die Memory Card auszuwerten. Alternative Methoden zum Neustarten der CPU sind ein Wechsel von STOP in RUN oder ein Urlöschen (MRES) in STEP 7.
3. Nach dem Neustart und der Auswertung der Memory Card kopiert die CPU das Programm in den internen Ladespeicher der CPU.

Die RUN/STOP-LED blinkt abwechselnd grün und gelb, um kenntlich zu machen, dass das Programm kopiert wird. Wenn die RUN/STOP-LED eingeschaltet wird (und dauerhaft gelb leuchtet) und die MAINT-LED blinkt, ist der Kopiervorgang beendet. Nun können Sie die Memory Card entnehmen.

4. Starten Sie die CPU neu (entweder durch Einschalten oder durch eine der alternativen Methoden zum Neustarten), um das neue Programm, das in den internen Ladespeicher übertragen wurde, auszuwerten.

Die CPU geht dann in den Betriebszustand für den Anlauf (RUN oder STOP), den Sie für das Projekt eingerichtet haben.

---

#### Hinweis

Sie müssen die Übertragungskarte ziehen, bevor Sie die CPU in RUN versetzen.

---

### 4.5.4 Einsatz der Memory Card als Programmkarte

#### VORSICHT

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.



Stellen Sie sicher, dass die Memory Card nicht schreibgeschützt ist. Schieben Sie dazu den Schutzschalter aus der Verriegelungsposition heraus.

Löschen Sie vor dem Kopieren von Programmelementen auf die Programmkarte alle zuvor gespeicherten Programme von der Memory Card.

## Programmkarte anlegen

Bei Einsatz als Programmkarte funktioniert die Memory Card als externer Ladespeicher der CPU. Wenn Sie die Programmkarte ziehen, ist der interne Ladespeicher der CPU leer.

### Hinweis

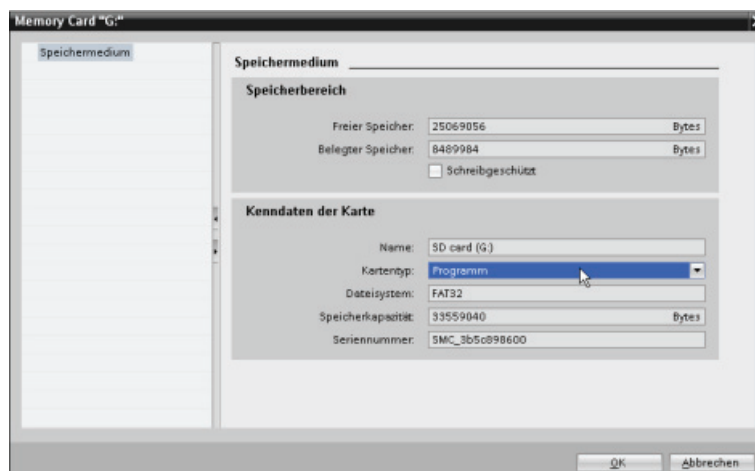
Wenn Sie eine leere Memory Card in die CPU stecken und eine Auswertung der Memory Card durchführen, indem Sie die CPU aus- und wieder einschalten, die CPU von STOP nach RUN versetzen oder ein Umlöschen (MRES) durchführen, werden das Programm und die geforderten Werte aus dem internen Ladespeicher der CPU in die Memory Card kopiert. (Die Memory Card ist nun eine Programmkarte.) Nach Abschluss des Kopiervorgangs wird das Programm im internen Ladespeicher der CPU gelöscht. Die CPU geht dann in den konfigurierten Betriebszustand für den Anlauf (RUN oder STOP).

Denken Sie daran, die Anlaufparameter der CPU zu konfigurieren (Seite 119), bevor Sie ein Projekt auf die Programmkarte kopieren. Um eine Programmkarte anzulegen, gehen Sie folgendermaßen vor:

1. Stecken Sie eine leere SIMATIC Memory Card in einen SD-Kartenleser, der an Ihren Computer angeschlossen ist.


Wenn Sie eine SIMATIC Memory Card verwenden, die bereits ein Anwenderprogramm oder ein Firmware-Update enthält, **müssen** Sie die Programmdateien löschen, bevor Sie die Karte erneut verwenden. Rufen Sie den Windows Explorer auf und zeigen Sie die Inhalte der Memory Card an. Löschen Sie die Datei "S7\_JOB.S7S" sowie alle vorhandenen Ordner mit Datenprotokollen und alle Verzeichnisordner (wie "SIMATIC.S7S" oder "FWUPDATE.S7S").

2. Erweitern Sie in der Projektnavigation (Projektansicht) den Ordner "SIMATIC Kartenleser" und wählen Sie Ihren Kartenleser aus.
3. Klicken Sie mit der rechten Maustaste auf den Laufwerksbuchstaben der Memory Card im Kartenleser und wählen Sie im Kontextmenü die Option "Eigenschaften". Daraufhin wird der Dialog "Memory Card" angezeigt.
4. Wählen Sie im Dialog "Memory Card" in der Klappliste die Option "Programm" aus.



5. Fügen Sie das Programm hinzu, indem Sie in der Projektnavigation die CPU (z.B. PLC\_1 [CPU 1214 DC/DC/DC]) auswählen und mit der Maus auf die Memory Card ziehen. (Alternativ können Sie die CPU kopieren und in die Memory Card einfügen.) Durch Kopieren der CPU in die Memory Card wird der Dialog "Vorschau laden" geöffnet.
6. Klicken Sie im Dialog "Vorschau laden" auf die Schaltfläche "Laden", um die CPU in die Memory Card zu kopieren.
7. Wenn der Dialog eine Meldung anzeigt, dass die CPU (das Programm) fehlerfrei geladen wurde, klicken Sie auf die Schaltfläche "Fertig stellen".

### Programmkarte als externen Ladespeicher für die CPU nutzen

 <b>WARNUNG</b>
<b>Vergewissern Sie sich vor dem Einlegen der Memory Card, dass die CPU keinen Prozess ausführt.</b>
Durch das Einlegen einer Memory Card geht die CPU in den Betriebszustand STOP, was sich auf den Betrieb eines Online-Prozesses oder einer Maschine auswirken kann. Unvorhersehbarer Betrieb eines Prozesses oder einer Maschine kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.
Stellen Sie vor dem Einlegen einer Memory Card sicher, dass die CPU offline und in einem sicheren Zustand ist.

Um mit Ihrer CPU eine Programmkarte zu nutzen, gehen Sie folgendermaßen vor:

1. Schieben Sie die Programmkarte in die CPU. Befindet sich die CPU im Betriebszustand RUN, geht sie in STOP. Die Wartungs-LED (MAINT) blinkt, um anzuzeigen, dass die Memory Card ausgewertet werden muss.
2. Schalten Sie die CPU aus und wieder ein, um die Memory Card auszuwerten. Alternative Methoden zum Neustarten der CPU sind ein Wechsel von STOP in RUN oder ein Urlöschen (MRES) in STEP 7.
3. Nach dem Neustart der CPU und der Auswertung der Programmkarte löscht die CPU den internen Ladespeicher der CPU.

Die CPU geht dann in den Betriebszustand für den Anlauf (RUN oder STOP), den Sie für die CPU eingerichtet haben.

Die Programmkarte muss in der CPU gesteckt bleiben. Wenn Sie die Programmkarte ziehen, hat die CPU kein Programm mehr im internen Ladespeicher.

 **WARNUNG**

Wenn Sie die Programmkarte ziehen, verliert die CPU den externen Ladespeicher und erzeugt einen Fehler. Die CPU geht dann in den Betriebszustand STOP und die Fehler-LED blinkt.

Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch den unkontrollierten Betrieb der gesteuerten Geräte verursachen. Daraus resultiert ein unvorhersehbarer Betrieb des Automatisierungssystems, der zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.

### 4.5.5 Firmware-Update

**VORSICHT**

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.

Sie verwenden die Memory Card auch zum Herunterladen von Firmware-Updates vom Kunden-Support (<http://www.siemens.com/automation/>). Wählen Sie auf dieser Website den Befehl **Automatisierungstechnik > Automatisierungssysteme > Industrie-Automatisierungssysteme SIMATIC > Steuerungen > Modulare Controller SIMATIC S7 > SIMATIC S7-1200**. Von dort navigieren Sie zu dem speziellen Modultyp, den Sie aktualisieren möchten. Klicken Sie unter "Support" auf den Link für "Software Downloads", um fortzufahren.

Alternativ können Sie die S7-1200 Website für Downloads (<http://support.automation.siemens.com/WW/view/de/34612486/133100>) direkt aufrufen.

---

**Hinweis**

Sie können eine S7-1200 CPU V2.2 oder eine Vorgängerversion davon nicht auf die Firmware S7-1200 V3.0 aktualisieren.

---

**VORSICHT**

**Verwenden Sie zum Neuformatieren der Memory Card nicht das Windows-Formatierungsprogramm oder ein anderes Formatierungsprogramm.**

Wenn eine Siemens Memory Card mit dem Windows-Formatierungsprogramm neu formatiert wird, kann die Memory Card nicht mehr von einer S7-1200 CPU verwendet werden.

Um das Firmware-Update auf Ihre Memory Card zu laden, gehen Sie folgendermaßen vor:

1. Stecken Sie eine leere SIMATIC MC 24-MB-Memory-Card in einen SD-Kartenleser, der an Ihren Computer angeschlossen ist.

Sie können eine SIMATIC Memory Card verwenden, die bereits ein Anwenderprogramm oder ein anderes Firmware-Update enthält, doch Sie müssen einige der Dateien auf der Memory Card löschen.

**VORSICHT**

**Löschen Sie NICHT die versteckten Dateien "\_\_LOG\_\_" und "crdinfo.bin" von der Memory Card.**

Die Dateien "\_\_LOG\_\_" und "crdinfo.bin" werden von der Memory Card benötigt. Wenn Sie diese Dateien löschen, können Sie die Memory Card nicht mehr mit der CPU nutzen.

Um eine Memory Card wiederzuverwenden, **müssen** Sie die Datei "S7\_JOB.S7S" sowie alle vorhandenen Ordner mit Datenprotokollen und alle Verzeichnisordner (wie "SIMATIC.S7S" oder "FWUPDATE.S7S") löschen, bevor Sie das Firmware-Update herunterladen. Löschen Sie jedoch **nicht** die Dateien "\_\_LOG\_\_" und "crdinfo.bin". (Diese Dateien sind typischerweise versteckt und werden benötigt.) Mit dem Windows Explorer können Sie den Inhalt der Memory Card anzeigen und die Datei und Ordner löschen.

2. Wählen Sie die selbst extrahierende Datei (\*.exe) des Ihrem Modul entsprechenden Firmware-Updates aus und laden Sie sie auf Ihren Computer herunter. Doppelklicken Sie auf die Update-Datei, geben Sie als Zielpfad das Stammverzeichnis der SIMATIC Memory Card an und starten Sie die Extraktion. Wenn die Extraktion beendet ist, enthält das Stammverzeichnis (Ordner) der Memory Card das Verzeichnis "FWUPDATE.S7S" und die Datei "S7\_JOB.S7S".

Zum Installieren des Firmware-Updates gehen Sie folgendermaßen vor:

 <b>WARNUNG</b>
<b>Stellen Sie vor der Installation des Firmware-Updates sicher, dass die CPU keinen aktiven Prozess ausführt.</b>
Durch die Installation des Firmware-Updates geht die CPU in den Betriebszustand STOP, was sich auf den Betrieb eines Online-Prozesses oder einer Maschine auswirken kann. Unerwarteter Betrieb eines Prozesses oder einer Maschine kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.
Bevor Sie die Memory Card stecken, stellen Sie unbedingt sicher, dass die CPU offline und in einem sicheren Zustand ist.

1. Schieben Sie die Memory Card in die CPU. Befindet sich die CPU im Betriebszustand RUN, geht sie in STOP. Die Wartungs-LED (MAINT) blinkt, um anzuzeigen, dass die Memory Card ausgewertet werden muss.
2. Starten Sie die CPU neu, um das Firmware-Update zu starten. Alternative Methoden zum Neustarten der CPU sind ein Wechsel von STOP in RUN oder ein Urlöschen (MRES) in STEP 7.

<b>ACHTUNG</b>
Damit das Firmware-Upgrade für das Modul abgeschlossen werden kann, achten Sie darauf, dass die externe 24-V-DC-Versorgung des Moduls eingeschaltet bleibt.

Nach dem Neustart der CPU beginnt das Firmware-Update. Die RUN/STOP-LED blinkt grün und gelb, um anzuzeigen, dass das Update kopiert wird. Wenn die RUN/STOP-LED dauerhaft gelb leuchtet und die MAINT-LED blinkt, ist der Kopiervorgang beendet. Anschließend müssen Sie die Memory Card entnehmen.

3. Nachdem Sie die Memory Card entnommen haben, starten Sie die CPU erneut neu (entweder durch Einschalten oder mit einem anderen Verfahren zum Neustarten), um die neue Firmware zu laden.

Das Anwenderprogramm und die Hardware-Konfiguration sind vom Firmware-Update nicht betroffen. Wenn die CPU eingeschaltet wird, geht die CPU in den konfigurierten Anlaufzustand. (Wenn als Anlaufbetriebszustand für die CPU "Warmstart - Betriebsart vor NETZ-AUS" konfiguriert wurde, befindet sich die CPU im Betriebszustand STOP, weil STOP der letzte Betriebszustand der CPU war.)

---

**Hinweis****Mehrere an eine CPU angeschlossene Module aktualisieren**

Wenn Ihre Hardwarekonfiguration mehrere Module enthält, auf die ein Firmware-Update von der Memory Card angewendet werden kann, aktualisiert die CPU alle betroffenen Module (CM, SM, SB) in der Konfigurationsreihenfolge, d.h. in aufsteigender Reihenfolge der Modulposition in der Gerätekonfiguration in STEP 7.

Wenn Sie mehrere Firmware-Updates für mehrere Module auf die Memory Card geladen haben, führt die CPU die Updates in der Reihenfolge durch, in der Sie sie auf die Memory Card geladen haben.

---

## 4.6 Vorgehensweise bei verlorenem Passwort

Wenn Sie das Passwort für eine passwortgeschützte CPU verloren haben, löschen Sie das passwortgeschützte Programm mit einer leeren Übertragungskarte. Die leere Übertragungskarte löscht den internen Ladespeicher der CPU. Dann können Sie ein neues Anwenderprogramm aus STEP 7 in die CPU laden.

Informationen zum Erstellen und Nutzen einer leeren Übertragungskarte finden Sie im Abschnitt Übertragungskarten (Seite 119).

 <b>WARNUNG</b>
--

Wenn Sie eine Übertragungskarte in eine laufende CPU stecken, geht die CPU in STOP. Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch den unkontrollierten Betrieb der gesteuerten Geräte verursachen. Daraus resultiert ein unvorhersehbarer Betrieb des Automatisierungssystems, der zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.
--

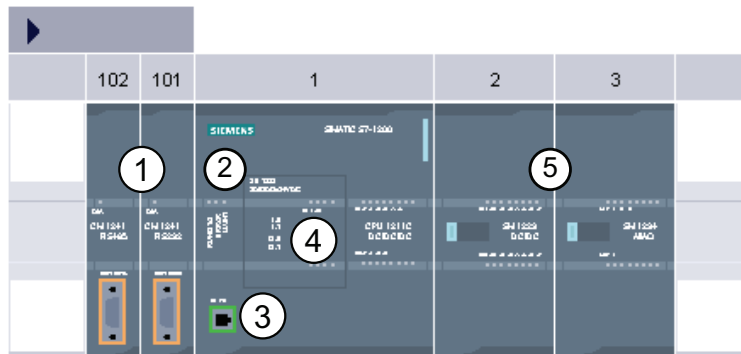
Sie müssen die Übertragungskarte ziehen, bevor Sie die CPU in RUN versetzen.





## Gerätekonfiguration

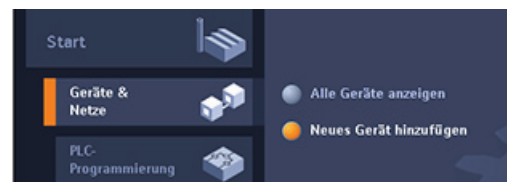
Sie können die Gerätekonfiguration für Ihr PLC-Gerät durch Hinzufügen einer CPU und weiterer Module zu Ihrem Projekt erstellen.



- ① Kommunikationsmodul (CM) oder Kommunikationsprozessor (CP): bis zu 3, in Steckplätzen 101, 102 und 103
- ② CPU: Steckplatz 1
- ③ Ethernet-Anschluss der CPU
- ④ Signalboard (SB), Kommunikationsboard (CB) oder Batterieboard (BB): max. 1, in CPU gesteckt
- ⑤ Signalmodul (SM) für digitale oder analoge E/A: bis zu 8, in Steckplätzen 2 bis 9 (8 bei der CPU 1214C und CPU 1215C, 2 bei der CPU 1212C, keins bei der CPU 1211C)

Um die Gerätekonfiguration anzulegen, fügen Sie Ihrem Projekt zunächst ein Gerät hinzu.

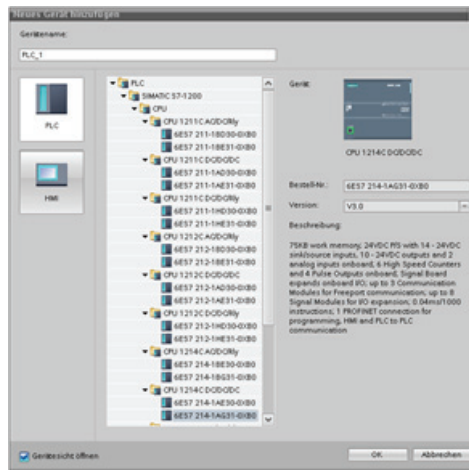
- Wählen Sie in der Portalansicht das Portal "Geräte & Netze" und klicken Sie auf "Neues Gerät hinzufügen".
- Doppelklicken Sie in der Projektansicht unter dem Projektamen auf "Neues Gerät hinzufügen".



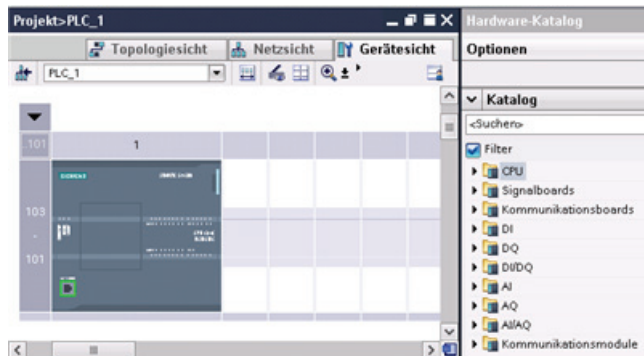
## 5.1 CPU einfügen

Zum Erstellen Ihrer Gerätekonfiguration fügen Sie eine CPU in Ihr Projekt ein. Achten Sie darauf, dass Sie die richtige Variante und Firmware-Version aus der Liste auswählen. Durch die Auswahl der CPU im Dialog "Neues Gerät hinzufügen" werden der Baugruppenträger und die CPU ausgewählt.

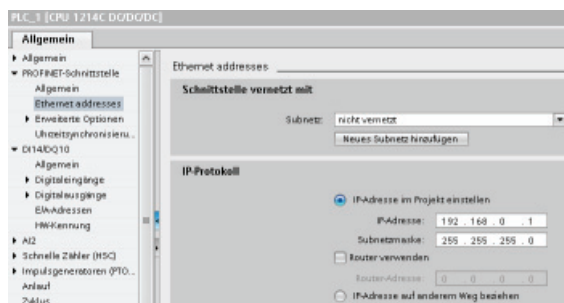
Dialog "Neues Gerät hinzufügen"



Gerätesicht der Hardwarekonfiguration



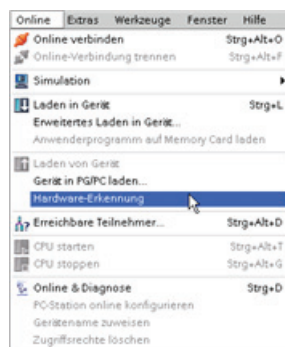
Nach der Auswahl der CPU in der Gerätesicht werden die Eigenschaften der CPU im Inspektorfenster angezeigt.



**Hinweis**

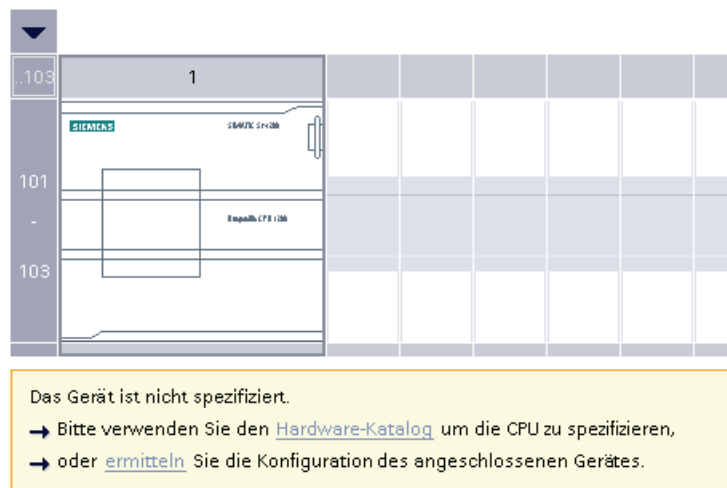
Die CPU hat keine vorkonfigurierte IP-Adresse. Sie müssen der CPU daher bei der Gerätekonfiguration manuell eine IP-Adresse zuweisen. Ist Ihre CPU an einen Router im Netzwerk angeschlossen, so muss auch die IP-Adresse des Routers eingegeben werden.

## 5.2 Konfiguration für eine nicht spezifizierte CPU erkennen

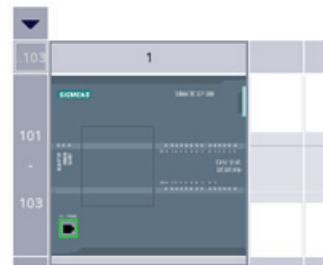
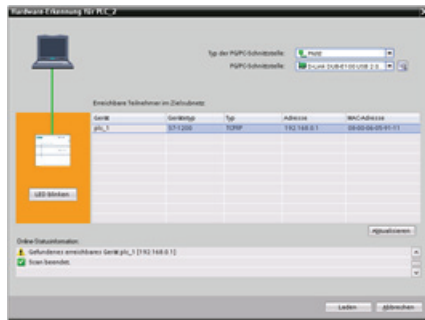


Wenn eine Verbindung zu einer CPU besteht, können Sie die Konfiguration dieser CPU einschließlich evtl. vorhandener Module aus dem Gerät in Ihr Projekt laden. Legen Sie dazu einfach ein neues Projekt an und wählen Sie anstelle einer bestimmten CPU die "nicht spezifizierte CPU". (Sie können auch die Gerätekonfiguration ganz umgehen, indem Sie unter "Erste Schritte" auf "Ein PLC-Programm erstellen" klicken. STEP 7 legt dann automatisch eine nicht spezifizierte CPU an.) Wählen Sie im Programmiereditor im Menü "Online" den Befehl "Hardwareerkennung".

Wählen Sie im Gerätekonfigurationseditor die Option zum Erkennen der Konfiguration des angeschlossenen Geräts.



Nachdem Sie im Online-Dialog die CPU ausgewählt und auf die Schaltfläche zum Laden geklickt haben, lädt STEP 7 die Hardwarekonfiguration einschließlich möglicher Module (SM, SB oder CM) aus der CPU. Sie können dann die Parameter für die CPU und die Module konfigurieren.




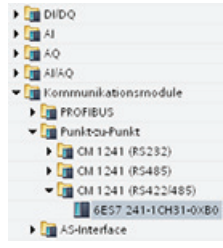
## 5.3 Module zur Konfiguration hinzufügen

Im Hardwarekatalog können Sie Module zur CPU hinzufügen:

- Signalmodule (SMs) für zusätzliche digitale oder analoge Ein- und Ausgänge. Diese Module werden an der rechten Seite der CPU angeschlossen.
- Signalboards (SBs) bieten eine begrenzte Zahl von zusätzlichen Ein-/Ausgängen für die CPU. Das SB wird auf der Vorderseite der CPU gesteckt.
- Das Batterieboard 1297 (BB) bietet eine Langzeitpufferung der Echtzeituhr. Das BB wird auf der Vorderseite der CPU gesteckt.
- Kommunikationsboards (CBs) bieten einen zusätzlichen Kommunikationsanschluss (z. B. RS485). Das CB wird auf der Vorderseite der CPU gesteckt.
- Kommunikationsmodule (CMs) und Kommunikationsprozessoren (CPs) bieten einen zusätzlichen Kommunikationsanschluss, z. B. für PROFIBUS oder GPRS. Diese Module werden an der linken Seite der CPU angeschlossen.

Um ein Modul in die Gerätekonfiguration einzufügen, wählen Sie das Modul im Hardwarekatalog aus und klicken doppelt darauf oder ziehen Sie es in den markierten Steckplatz. Sie müssen die Module in die Gerätekonfiguration aufnehmen und die Hardwarekonfiguration in die CPU laden, damit die Module funktionsfähig sind.

Tabelle 5- 1 Modul zur Gerätekonfiguration hinzufügen

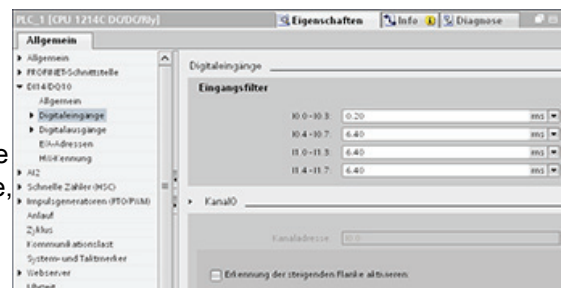
Modul	Modul auswählen	Modul einsetzen	Ergebnis
SM			
SB, BB oder CB			
CM oder CP			

## 5.4 Konfigurieren des CPU-Betriebs

Um die Betriebsparameter der CPU zu konfigurieren, wählen Sie die CPU in der Gerätesicht aus (blauer Rahmen um die gesamte CPU) und öffnen dann im Inspektorfenster das Register "Eigenschaften".

Um die Zeiten für den Eingangsfiler einzurichten, wählen Sie "Digitaleingänge" aus. Die Standardfilterzeit für die digitalen Eingänge beträgt 6,4 ms.

Jeder Eingang hat eine Filterkonfiguration, die für alle Zwecke gilt: Prozesseingänge, Alarme, Impulsabgriff und HSC-Eingänge.



**Hinweis**

Wenn für einen HSC kein Eingang konfiguriert ist, gilt die in diesem Dialogfeld ausgewählte Filtereinstellung. Wenn für einen HSC ein Eingang konfiguriert ist, sind für die Filtereinstellung dieses Eingangs automatisch 800 ns festgelegt, und dieser Wert wird von der Konfiguration in diesem Dialogfeld nicht beeinflusst.


 <b>WARNUNG</b>
<p>Wenn die bisherige Einstellung der Filterzeit eines Digitaleingangskanals verändert wird, muss möglicherweise bis zu 20,0 ms lang ein neuer Eingangswert mit Pegel "0" vorhanden sein, damit der Filter vollständig auf neue Eingänge reagiert. Während dieses Zeitraums werden kurze Impulse mit Pegel "0", die kürzer als 20,0 ms sind, möglicherweise nicht erkannt oder gezählt.</p> <p>Das Verändern der Filterzeiten kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.</p> <p>Um sicherzustellen, dass eine neue Filterzeit sofort wirksam wird, müssen Sie die CPU aus- und wieder einschalten.</p>

Tabelle 5- 2 CPU-Eigenschaften

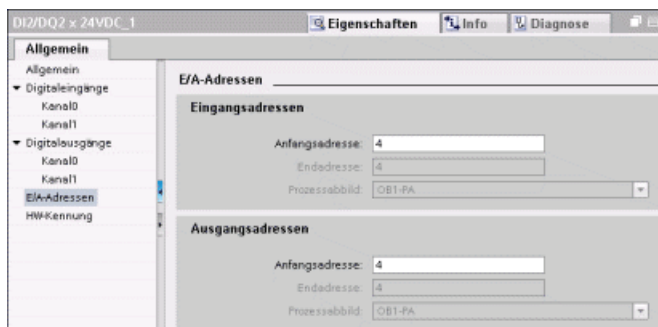
Eigenschaft	Beschreibung
PROFINET-Schnittstelle	Einstellen der IP-Adresse für die CPU und die Uhrzeitsynchronisation
DE, DA und AE	Einstellen des Verhaltens der lokalen (integrierten) digitalen und analogen E/A (z.B. Filterzeiten für die Digitaleingänge und die Reaktion von Digitalausgängen auf das Versetzen der CPU in den Betriebszustand STOP).
Schnelle Zähler (Seite 357) und Impulsgeneratoren (Seite 330)	Aktivieren und Einstellen der schnellen Zähler (HSC) und der Impulsgeneratoren für die Impulsfolgen (PTO) und die Impulsdauermodulation (PWM) Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM oder Bewegungssteuerungsanweisungen) konfigurieren, werden die entsprechenden Adressen der Ausgänge (A0.0, A0.1, A4.0 und A4.1) aus dem Speicher der Ausgänge entfernt und können nicht für andere Zwecke in Ihrem Anwenderprogramm verwendet werden. Wenn Ihr Anwenderprogramm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.
Anlauf (Seite 75)	<b>Anlauf nach NETZ-EIN:</b> Einstellen des Verhaltens der CPU nach dem Aus- und Wiedereinschalten, z. B. für das Anlaufen im Betriebszustand STOP oder den Wechsel in RUN nach einem Warmstart

Eigenschaft	Beschreibung
	<p><b>Hardwarekompatibilität:</b> Konfiguration der Ersatzstrategie für alle Systemkomponenten (SM, SB, CM, CP und CPU):</p> <ul style="list-style-type: none"> <li>• Akzeptablen Ersatz zulassen</li> <li>• Beliebigen Ersatz zulassen (Standard)</li> </ul> <p>Jedes Modul enthält interne Anforderungen an die Ersatzkompatibilität basierend auf der Anzahl der E/A, der elektrischen Verträglichkeit und anderen entsprechenden Vergleichspunkten. Beispiel: Ein SM mit 16 Kanälen kann ein akzeptabler Ersatz für ein SM mit 8 Kanälen sein, doch ein SM mit 8 Kanälen kann kein akzeptabler Ersatz für ein SM mit 16 Kanälen sein. Wenn Sie "Akzeptablen Ersatz zulassen" wählen, setzt STEP 7 die Ersetzungsregeln um. Ansonsten gestattet STEP 7 jeden Ersatz.</p> <p><b>Zuweisungszeit der Parameter für dezentrale E/A:</b> Legt einen maximalen Zeitraum (Voreinstellung: 60000 ms) fest, in dem die dezentrale Peripherie online gebracht werden muss. (Die CMs und CPs werden während des Anlaufs von der CPU mit Spannung und Kommunikationsparametern versorgt. Diese Zuweisungszeit gestattet einen Zeitraum, während dessen die an CM oder CP angeschlossenen E/A-Module online gebracht werden müssen.)</p> <p>Die CPU geht in RUN, sobald die dezentrale Peripherie online ist, unabhängig von der Zuweisungszeit. Wurde die dezentrale Peripherie während dieses Zeitraums nicht online gebracht, geht die CPU trotzdem in RUN, ohne die dezentrale Peripherie.</p> <p><b>Hinweis:</b> Wenn Ihre Konfiguration ein CM 1243-5 (PROFIBUS-Master) enthält, legen Sie für diesen Parameter keinen Wert unter 15 Sekunden (15000 ms) fest, um sicherzustellen, dass das Modul online gebracht wird.</p>
Zyklus (Seite 87)	Festlegen einer maximalen Zykluszeit oder einer festen Mindestzykluszeit
Kommunikationslast	Zuweisen eines prozentualen Anteils der CPU-Zeit für Kommunikationsaufgaben
System- und Taktmerker (Seite 92)	Setzen eines Bytes für die Systemmerkerfunktionen und Setzen eines Bytes für die Taktmerkerfunktionen (dabei schaltet jedes Bit in einer vorgegebenen Frequenz ein und aus).
Webserver (Seite 531)	Aktiviert und konfiguriert die Webserverfunktion.
Tageszeit	Auswählen der Zeitzone und Einstellen der Sommer-/Winterzeit
Schutz (Seite 176)	Einstellen des Lese-/Schreibschutzes und des Passworts für den Zugriff auf die CPU
Verbindungsressourcen (Seite 451)	Bietet einen Überblick über die Kommunikationsverbindungen der CPU und die Anzahl der konfigurierten Verbindungen.
Übersicht über Adressen	Bietet einen Überblick über die E/A-Adressen, die für die CPU konfiguriert sind.

## 5.5 Modulparameter konfigurieren

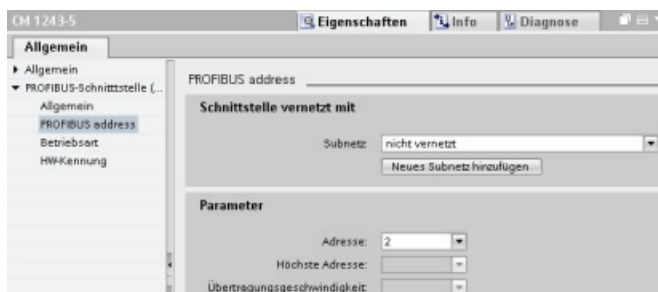
Um die Betriebsparameter der Module zu konfigurieren, wählen Sie das Modul in der Gerätesicht aus und öffnen im Inspektorfenster das Register "Eigenschaften", um die Parameter für das Modul einzurichten.

### Signalmodul (SM) oder Signalboard (SB) konfigurieren



- Digitale E/A: Die Eingänge können für die Erkennung steigender oder fallender Flanken (jeweils mit Zuweisung eines Ereignisses und eines Prozessalarms) und auch für den "Impulsabgriff" (Eingang bleibt nach einem Impuls eingeschaltet) durch die nächste Aktualisierung des Prozessabbilds der Eingänge konfiguriert werden. Die Ausgänge können eingefroren werden oder es können Ersatzwerte aufgeschaltet werden.
- Analoge E/A: Für einzelne Eingänge konfigurieren Sie Parameter, wie z.B. Messart (Spannung oder Strom), Bereich und Glättung sowie Freigabe der Unter- oder Überlaufdiagnose. Die Analogausgänge bieten Parameter wie Ausgangsart (Spannung oder Strom) und Diagnose, wie z. B. Kurzschluss (bei Spannungsausgängen) oder Diagnose der oberen/unteren Grenzwerte. Die Bereiche von Analogeingängen und Analogausgängen in physikalischen Einheiten konfigurieren Sie nicht im Dialog "Eigenschaften". Dies nehmen Sie in Ihrer Programmlogik vor (siehe "Verarbeitung von Analogwerten (Seite 101)").
- E/A-Diagnoseadressen: Konfigurieren der Startadressen für die Ein- und Ausgänge des Moduls

### Kommunikationsschnittstelle (CM, CP oder CB) konfigurieren



Sie konfigurieren die Parameter für das Netzwerk je nach Typ der Kommunikationsschnittstelle.






## 5.6 CPU für die Kommunikation konfigurieren

### 5.6.1 Netzwerkverbindung erstellen

In der Netzansicht der Gerätekonfiguration können Sie die Netzwerkverbindungen zwischen den Geräten in Ihrem Projekt herstellen. Nach dem Herstellen der Netzwerkverbindung können Sie im Register "Eigenschaften" des Inspektorfensters die Netzwerkparameter konfigurieren.

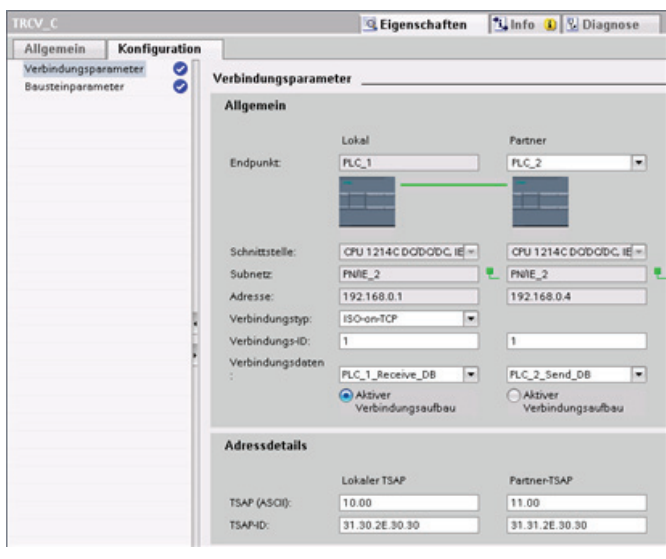
Tabelle 5- 3 Netzwerkverbindung erstellen

Handlungsanweisung	Ergebnis
Wählen Sie die "Netzwerkansicht" für die Anzeige der Geräte, die verbunden werden sollen.	
Wählen Sie die Schnittstelle eines Geräts und ziehen Sie die Verbindung zur Schnittstelle des zweiten Geräts.	
Lassen Sie die Maustaste los, damit die Netzwerkverbindung erstellt wird.	

### 5.6.2 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

Im Inspektorfenster werden die Eigenschaften der Verbindung angezeigt, wenn Sie einen Teil der Anweisung auswählen. Sie legen die Kommunikationsparameter über das Register "Konfiguration" im Dialog "Eigenschaften" der Kommunikationsanweisung fest.

Tabelle 5- 4 Verbindungspfad konfigurieren (über die Eigenschaften der Anweisung)

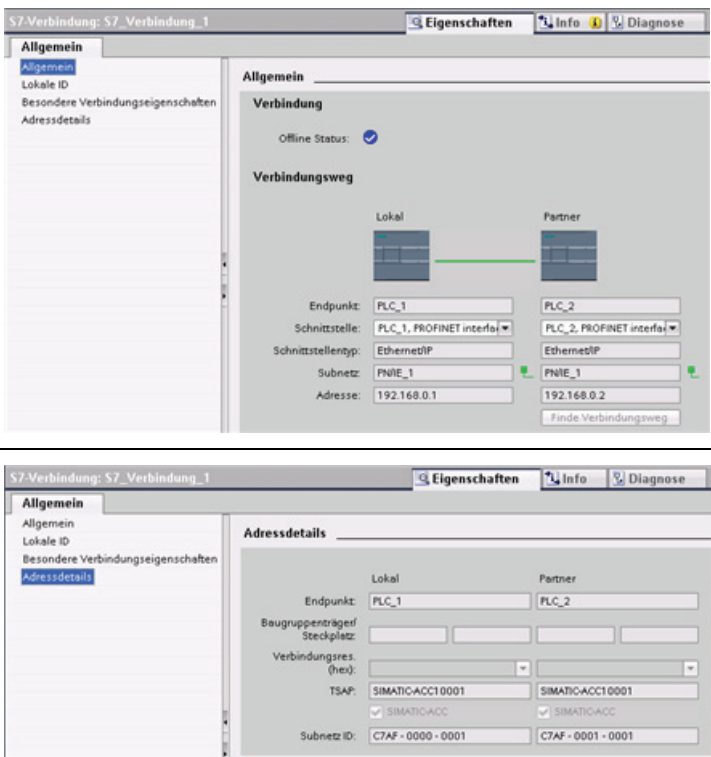
TCP, ISO-on-TCP und UDP	Verbindungseigenschaften
<p>Bei den TCP-, ISO-on-TCP- und UDP-Ethernet-Protokollen konfigurieren Sie die Verbindungen der lokalen und der Partner-CPU in den "Eigenschaften" der Anweisung (TSEND_C, TRCV_C oder TCON).</p> <p>Die Abbildung zeigt die "Verbindungseigenschaften" im Register "Verbindung" einer ISO-on-TCP-Verbindung.</p>	

#### Hinweis

Wenn Sie die Verbindungseigenschaften für eine CPU konfigurieren, können Sie in STEP 7 entweder einen bestimmten Verbindungs-DB in der Partner-CPU auswählen (sofern vorhanden) oder einen Verbindungs-DB für die Partner-CPU anlegen. Die Partner-CPU muss im Projekt bereits angelegt sein, es darf sich nicht um eine "nicht spezifizierte" CPU handeln.

Sie müssen trotzdem eine Anweisung TSEND\_C, TRCV\_C oder TCON in das Anwenderprogramm der Partner-CPU einfügen. Wenn Sie die Anweisung einfügen, wählen Sie den Verbindungs-DB aus, der von der Konfiguration angelegt wurde.

Tabelle 5- 5 Verbindungspfad für die S7-Kommunikation konfigurieren (Gerätekonfiguration)

S7-Kommunikation (GET und PUT)	Verbindungseigenschaften
<p>Bei der S7-Kommunikation konfigurieren Sie die Verbindungen zwischen lokaler und Partner-CPU im Editor "Geräte &amp; Netze" des Netzwerks. Sie können auf die Schaltfläche "Hervorgehoben: Verbindung" klicken, um die "Eigenschaften" aufzurufen.</p> <p>Im Register "Allgemein" werden mehrere Eigenschaften aufgeführt:</p> <ul style="list-style-type: none"> <li>• "Allgemein" (abgebildet)</li> <li>• "Lokale ID"</li> <li>• "Spezielle Verbindungseigenschaften"</li> <li>• "Adressdetails" (abgebildet)</li> </ul>	

Im Abschnitt "PROFINET" unter "Protokolle" (Seite 457) und im Abschnitt "S7-Kommunikation" unter "S7-Verbindung erstellen" (Seite 522) finden Sie weitere Informationen sowie eine Liste der verfügbaren Kommunikationsanweisungen.

Tabelle 5- 6 Parameter für die CPU-Verbindung

Parameter	Definition																
Adresse	Zugewiesene IP-Adressen																
Allgemein	<table border="1"> <tr> <td>Endpunkt</td> <td>Name der Partner-CPU (Empfänger)</td> </tr> <tr> <td>Schnittstelle</td> <td>Name der Schnittstellen</td> </tr> <tr> <td>Subnetz</td> <td>Name der Subnetze</td> </tr> <tr> <td>Schnittstellentyp</td> <td><i>Nur S7-Kommunikation:</i> Typ der Schnittstelle</td> </tr> <tr> <td>Verbindungstyp</td> <td>Typ des Ethernet-Protokolls</td> </tr> <tr> <td>Verbindungs-ID</td> <td>ID-Nummer</td> </tr> <tr> <td>Verbindungsdaten</td> <td>Datenspeicher für die lokale CPU und die Partner-CPU</td> </tr> <tr> <td>Aktive Verbindung herstellen</td> <td>Optionsfeld zum Auswählen der lokalen CPU oder der Partner-CPU als aktive Verbindung</td> </tr> </table>	Endpunkt	Name der Partner-CPU (Empfänger)	Schnittstelle	Name der Schnittstellen	Subnetz	Name der Subnetze	Schnittstellentyp	<i>Nur S7-Kommunikation:</i> Typ der Schnittstelle	Verbindungstyp	Typ des Ethernet-Protokolls	Verbindungs-ID	ID-Nummer	Verbindungsdaten	Datenspeicher für die lokale CPU und die Partner-CPU	Aktive Verbindung herstellen	Optionsfeld zum Auswählen der lokalen CPU oder der Partner-CPU als aktive Verbindung
Endpunkt	Name der Partner-CPU (Empfänger)																
Schnittstelle	Name der Schnittstellen																
Subnetz	Name der Subnetze																
Schnittstellentyp	<i>Nur S7-Kommunikation:</i> Typ der Schnittstelle																
Verbindungstyp	Typ des Ethernet-Protokolls																
Verbindungs-ID	ID-Nummer																
Verbindungsdaten	Datenspeicher für die lokale CPU und die Partner-CPU																
Aktive Verbindung herstellen	Optionsfeld zum Auswählen der lokalen CPU oder der Partner-CPU als aktive Verbindung																
Adressdetails	<table border="1"> <tr> <td>Endpunkt</td> <td><i>Nur S7-Kommunikation:</i> Name der Partner-CPU (Empfänger)</td> </tr> <tr> <td>Baugruppenträger/Steckplatz</td> <td><i>Nur S7-Kommunikation:</i> Baugruppenträger- und Steckplatzadresse</td> </tr> <tr> <td>Verbindungsressource</td> <td><i>Nur S7-Kommunikation:</i> Komponente des TSAP für die Konfiguration einer S7-Verbindung mit einer S7-300 oder S7-400 CPU</td> </tr> </table>	Endpunkt	<i>Nur S7-Kommunikation:</i> Name der Partner-CPU (Empfänger)	Baugruppenträger/Steckplatz	<i>Nur S7-Kommunikation:</i> Baugruppenträger- und Steckplatzadresse	Verbindungsressource	<i>Nur S7-Kommunikation:</i> Komponente des TSAP für die Konfiguration einer S7-Verbindung mit einer S7-300 oder S7-400 CPU										
Endpunkt	<i>Nur S7-Kommunikation:</i> Name der Partner-CPU (Empfänger)																
Baugruppenträger/Steckplatz	<i>Nur S7-Kommunikation:</i> Baugruppenträger- und Steckplatzadresse																
Verbindungsressource	<i>Nur S7-Kommunikation:</i> Komponente des TSAP für die Konfiguration einer S7-Verbindung mit einer S7-300 oder S7-400 CPU																

Parameter		Definition
	Port (dezimal):	TCP und UDP: Port der Partner-CPU im Dezimalformat
	TSAP <sup>1</sup> und Subnetz-ID:	ISO on TCP (RFC 1006) und S7-Kommunikation: TSAPs der lokalen CPU und der Partner-CPU im ASCII- und Hexadezimalformat

<sup>1</sup> Verwenden Sie beim Konfigurieren einer Verbindung mit einer S7-1200 CPU über ISO-on-TCP in der TSAP-Erweiterung für die passiven Kommunikationsteilnehmer nur ASCII-Zeichen.

### Transport Service Access Points (TSAPs)

Mit TSAPs gestatten das ISO-on-TCP-Protokoll und die S7-Kommunikation mehrere Verbindungen mit einer einzigen IP-Adresse (bis zu 64-K-Verbindungen). TSAPs ermitteln die eindeutige Zuordnung dieser Verbindungen der Kommunikationsendpunkte zu einer IP-Adresse.

Die zu verwendenden TSAPs definieren Sie im Dialog "Verbindungsparameter" unter "Adressdetails". Der TSAP einer Verbindung in der CPU wird im Feld "Lokaler TSAP" eingegeben. Der TSAP für die Verbindung in der Partner-CPU wird im Feld "Partner-TSAP" eingegeben.

### Portnummern

Bei den TCP- und UDP-Protokollen müssen in der Konfiguration der Verbindungsparameter der lokalen (aktiven) CPU die dezentrale IP-Adresse und die Portnummer der (passiven) Partner-CPU angegeben werden.

Die zu verwendenden Ports definieren Sie im Dialog "Verbindungsparameter" unter "Adressdetails". Der Port einer Verbindung in der CPU wird im Feld "Lokaler Port" eingegeben. Der Port für die Verbindung in der Partner-CPU wird im Feld "Partner-Port" eingegeben.

### 5.6.3 Parameter für die PROFINET-Verbindung

Bei den Anweisungen TSEND\_C, TRCV\_C und TCON müssen verbindungsbezogene Parameter angegeben werden, um eine Verbindung zum Partnergerät aufbauen zu können. Diese Parameter werden von der Struktur TCON\_Param für die TCP-, ISO-on-TCP- und UDP-Protokolle angegeben. Üblicherweise geben Sie diese Parameter in den "Eigenschaften" der Anweisung im Register "Konfiguration" an. Kann auf das Register "Konfiguration" nicht zugegriffen werden, müssen Sie die Struktur TCON\_Param programmatisch angeben.

Tabelle 5-7 Struktur der Verbindungsbeschreibung (TCON\_Param)

Byte	Parameter und Datentyp		Beschreibung
0 ... 1	block_length	UInt	Länge: 64 Bytes (fest)
2 ... 3	id	CONN_OUC (Word)	Referenz auf diese Verbindung: Wertebereich: 1 (Standard) bis 4095. Geben Sie den Wert dieses Parameters für die Anweisung TSEND_C, TRCV_C oder TCON unter ID an.

Byte	Parameter und Datentyp		Beschreibung
4	connection_type	USInt	Verbindungstyp: <ul style="list-style-type: none"> <li>• 17: TCP (Standard)</li> <li>• 18: ISO-on-TCP</li> <li>• 19: UDP</li> </ul>
5	active_est	Bool	ID der Verbindungsart: <ul style="list-style-type: none"> <li>• TCP und ISO-on-TCP: <ul style="list-style-type: none"> <li>– FALSCH: Passive Verbindung</li> <li>– WAHR: Aktive Verbindung (Standard)</li> </ul> </li> <li>• UDP: FALSCH</li> </ul>
6	local_device_id	USInt	ID der lokalen PROFINET- oder Industrial Ethernet-Schnittstelle: 1 (Standard)
7	local_tsap_id_len	USInt	Länge des Parameters local_tsap_id in Bytes; mögliche Werte: <ul style="list-style-type: none"> <li>• TCP: 0 (aktiv, Standard) oder 2 (passiv)</li> <li>• ISO-on-TCP: 2 bis 16</li> <li>• UDP: 2</li> </ul>
8	rem_subnet_id_len	USInt	Dieser Parameter wird nicht verwendet.
9	rem_staddr_len	USInt	Länge der Adresse des Partnerendpunkts in Bytes: <ul style="list-style-type: none"> <li>• 0: nicht angegeben (Parameter rem_staddr ist irrelevant)</li> <li>• 4 (Standard): Gültige IP-Adresse im Parameter rem_staddr (nur bei TCP und ISO-on-TCP)</li> </ul>
10	rem_tsap_id_len	USInt	Länge des Parameters rem_tsap_id in Bytes; mögliche Werte: <ul style="list-style-type: none"> <li>• TCP: 0 (passiv) oder 2 (aktiv, Standard)</li> <li>• ISO-on-TCP: 2 bis 16</li> <li>• UDP: 0</li> </ul>
11	next_staddr_len	USInt	Dieser Parameter wird nicht verwendet.
12 ... 27	local_tsap_id	Array [1..16] of Byte	Komponente der lokalen Adresse der Verbindung: <ul style="list-style-type: none"> <li>• TCP und ISO-on-TCP: lokale Port-Nr. (mögliche Werte: 1 bis 49151; empfohlene Werte: 2000...5000): <ul style="list-style-type: none"> <li>– local_tsap_id[1] = High Byte der Portnummer in Hexadezimalnotierung;</li> <li>– local_tsap_id[2] = Low Byte der Portnummer in Hexadezimalnotierung;</li> <li>– local_tsap_id[3-16] = irrelevant</li> </ul> </li> <li>• ISO-on-TCP: lokale TSAP-ID: <ul style="list-style-type: none"> <li>– local_tsap_id[1] = B#16#E0;</li> <li>– local_tsap_id[2] = Baugruppenträger und Steckplatz der lokalen Endpunkte (Bits 0 bis 4: Steckplatznummer, Bits 5 bis 7: Nummer des Baugruppenträgers);</li> <li>– local_tsap_id[3-16] = TSAP-Erweiterung, optional</li> </ul> </li> <li>• UDP: Dieser Parameter wird nicht verwendet.</li> </ul> Hinweis: Stellen Sie sicher, dass jeder Wert von local_tsap_id innerhalb der CPU eindeutig ist.

Byte	Parameter und Datentyp		Beschreibung
28 ... 33	rem_subnet_id	Array [1..6] of USInt	Dieser Parameter wird nicht verwendet.
34 ... 39	rem_staddr	Array [1..6] of USInt	Nur bei TCP und ISO-on-TCP: IP-Adresse des Partnerendpunkts. (Nicht relevant bei passiven Verbindungen.) Beispiel: Die IP-Adresse 192.168.002.003 wird in den folgenden Elementen des Arrays gespeichert: rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = irrelevant
40 ... 55	rem_tsap_id	Array [1..16] of Byte	Komponente der Partneradresse der Verbindung: <ul style="list-style-type: none"> <li>• TCP: Portnummer der Partner-CPU. Bereich: 1 bis 49151; empfohlene Werte: 2000 bis 5000): <ul style="list-style-type: none"> <li>– rem_tsap_id[1] = High Byte der Portnummer in Hexadezimalnotierung;</li> <li>– rem_tsap_id[2] = Low Byte der Portnummer in Hexadezimalnotierung;</li> <li>– rem_tsap_id[3-16] = irrelevant</li> </ul> </li> <li>• ISO-on-TCP: Partner-TSAP-ID: <ul style="list-style-type: none"> <li>– rem_tsap_id[1] = B#16#E0</li> <li>– rem_tsap_id[2] = Baugruppenträger und Steckplatz des Partnerendpunkts (Bits 0 bis 4: Steckplatznummer, Bits 5 bis 7: Nummer des Baugruppenträgers)</li> <li>– rem_tsap_id[3-16] = TSAP-Erweiterung, optional</li> </ul> </li> <li>• UDP: Dieser Parameter wird nicht verwendet.</li> </ul>
56 ... 61	next_staddr	Array [1..6] of Byte	Dieser Parameter wird nicht verwendet.
62 ... 63	spare	Word	Reserviert: W#16#0000

**Siehe auch**

Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 138)

## 5.6.4 IP-Adressen zuweisen

### 5.6.4.1 IP-Adressen zu Programmier- und Netzwerkgeräten zuweisen

Wenn Ihr Programmiergerät eine integrierte Adapterkarte nutzt, die an das LAN Ihrer Anlage (und möglicherweise an das Internet) angeschlossen ist, müssen die Netzwerk-ID der IP-Adresse und die Subnetzmaske Ihrer CPU mit der integrierten Adapterkarte des Programmiergeräts genau übereinstimmen. Die Netzwerk-ID ist der erste Teil der IP-Adresse (die ersten drei Oktette, z.B. **211.154.184.16**). Sie legt fest, welches IP-Netzwerk Sie nutzen. Die Subnetzmaske hat normalerweise den Wert **255.255.255.0**. Da sich Ihr Computer jedoch in einem Anlagen-LAN befindet, kann die Subnetzmaske andere Werte annehmen (z.B. **255.255.254.0**), um eindeutige Subnetze einzurichten. Die Subnetzmaske definiert in UND-Verknüpfung mit der IP-Adresse des Geräts die Grenzen eines IP-Subnetzes.

---

#### Hinweis

Im Internet, wo Ihre Programmiergeräte, Netzwerkgeräte und IP-Router mit der Welt kommunizieren, müssen eindeutige IP-Adressen zugeordnet werden, um Konflikte mit anderen Netzwerknutzern zu vermeiden. Wenden Sie sich an die IT-Abteilung Ihrer Firma, deren Mitarbeiter die Netzwerke in Ihrem Unternehmen kennen, um sich IP-Adressen zuweisen zu lassen.

---

Wenn Ihr Programmiergerät eine Ethernet/USB-Adapterkarte nutzt, die an ein einzelnes Netzwerk angeschlossen ist, müssen die Netzwerk-ID der IP-Adresse und die Subnetzmaske Ihrer CPU mit der Ethernet/USB-Adapterkarte des Programmiergeräts genau übereinstimmen. Die Netzwerk-ID ist der erste Teil der IP-Adresse (die ersten drei Oktette, z.B. **211.154.184.16**). Sie legt fest, welches IP-Netzwerk Sie nutzen. Die Subnetzmaske hat normalerweise den Wert **255.255.255.0**. Die Subnetzmaske definiert in UND-Verknüpfung mit der IP-Adresse des Geräts die Grenzen eines IP-Subnetzes.

---

#### Hinweis

Eine Ethernet/USB-Adapterkarte ist nützlich, wenn Sie Ihre CPU nicht an das Firmen-LAN anschließen möchten. Dieser Aufbau ist besonders während anfänglicher Tests bzw. während der Inbetriebnahmeprüfungen nützlich.

---

Tabelle 5- 8 Ethernet-Adressen zuweisen

Adapterkarte des Programmiergeräts	Netzwerktyp	IP-Adresse	Subnetzmaske
Integrierte Adapterkarte	Angeschlossen an Ihr Firmen-LAN (und möglicherweise das Internet)	Die Netzwerk-ID Ihrer CPU und der integrierten Adapterkarte des Programmiergeräts müssen exakt übereinstimmen. Die Netzwerk-ID ist der erste Teil der IP-Adresse (die ersten drei Oktette, z.B. <b>211.154.184</b> .16). Sie legt fest, welches IP-Netzwerk Sie nutzen.	Die Subnetzmaske Ihrer CPU und der integrierten Adapterkarte müssen exakt übereinstimmen. Die Subnetzmaske hat normalerweise den Wert <b>255.255.255.0</b> . Da sich Ihr Computer jedoch in einem Anlagen-LAN befindet, kann die Subnetzmaske andere Werte annehmen (z.B. <b>255.255.254.0</b> ), um eindeutige Subnetze einzurichten. Die Subnetzmaske definiert in UND-Verknüpfung mit der IP-Adresse des Geräts die Grenzen eines IP-Subnetzes.
Ethernet/USB-Adapterkarte	Angeschlossen an ein einzelnes Netzwerk	Die Netzwerk-ID Ihrer CPU und der Ethernet/USB-Adapterkarte des Programmiergeräts müssen exakt übereinstimmen. Die Netzwerk-ID ist der erste Teil der IP-Adresse (die ersten drei Oktette, z.B. <b>211.154.184</b> .16). Sie legt fest, welches IP-Netzwerk Sie nutzen.	Die Subnetzmaske Ihrer CPU und der Ethernet/USB-Adapterkarte müssen exakt übereinstimmen. Die Subnetzmaske hat normalerweise den Wert <b>255.255.255.0</b> . Die Subnetzmaske definiert in UND-Verknüpfung mit der IP-Adresse des Geräts die Grenzen eines IP-Subnetzes.

**IP-Adresse Ihres Programmiergeräts über "Netzwerkumgebung" (auf Ihrem Desktop) zuweisen oder prüfen**

Die IP-Adresse Ihres Programmiergeräts können Sie mit den folgenden Menübefehlen zuweisen oder prüfen:

- (Klicken Sie mit der rechten Maustaste auf) "Netzwerkumgebung".
- "Eigenschaften"
- Klicken Sie mit der rechten Maustaste auf "LAN-Verbindung".
- "Eigenschaften"

Blättern Sie im Dialog "Eigenschaften von LAN-Verbindung" im Feld "Diese Verbindung verwendet folgende Elemente:" zum Eintrag "Internetprotokoll (TCP/IP)". Klicken Sie auf "Internetprotokoll (TCP/IP)" und dann auf die Schaltfläche "Eigenschaften". Wählen Sie "IP-Adresse automatisch beziehen (DHCP)" oder "Folgende IP-Adresse verwenden" (um eine statische IP-Adresse einzugeben) aus.

**Hinweis**

Das Dynamic Host Configuration Protocol (DHCP) weist Ihrem Programmiergerät beim Einschalten des DHCP-Servers automatisch eine IP-Adresse zu.

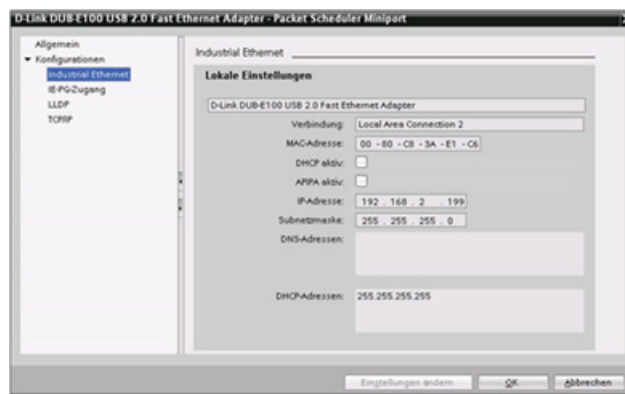


### 5.6.4.2 Ermitteln der IP-Adresse Ihres Programmiergeräts

Die MAC- und die IP-Adresse Ihres Programmiergeräts ermitteln Sie mit den folgenden Menübefehlen:

1. Erweitern Sie in der Projektnavigation den Knoten "Online-Zugänge".
2. Klicken Sie mit der rechten Maustaste auf das erforderliche Netzwerk und wählen Sie "Eigenschaften".
3. Erweitern Sie im Dialog "Netzwerk" die "Konfigurationen" und wählen Sie "Industrial Ethernet".

Dort werden die MAC- und die IP-Adresse Ihres Programmiergeräts angezeigt.

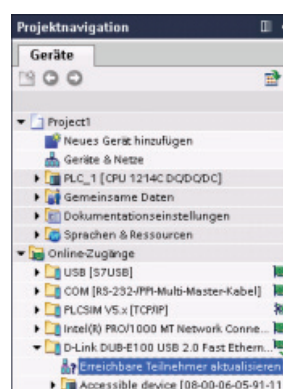


### 5.6.4.3 Online eine IP-Adresse zu einer CPU zuweisen

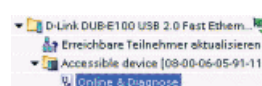
Sie können einem Netzwerkgerät online eine IP-Adresse zuweisen. Dies ist besonders bei der ersten Gerätekonfiguration nützlich.

1. Stellen Sie in der "Projektnavigation" mit folgenden Befehlen sicher, dass der CPU keine IP-Adresse zugewiesen ist:

- "Online-Zugänge"
- <Adapterkarte für das Netzwerk, in dem sich das Gerät befindet>
- "Erreichbare Teilnehmer aktualisieren"



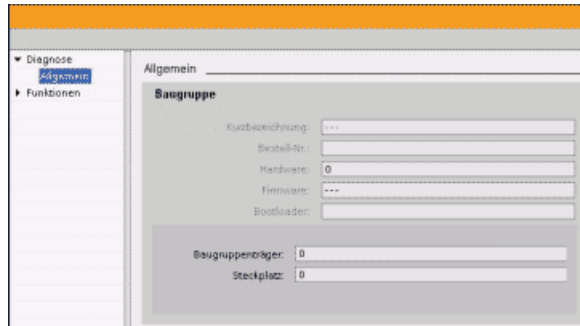
2. Doppelklicken Sie unter dem erforderlichen zugänglichen Gerät auf "Online & Diagnose".



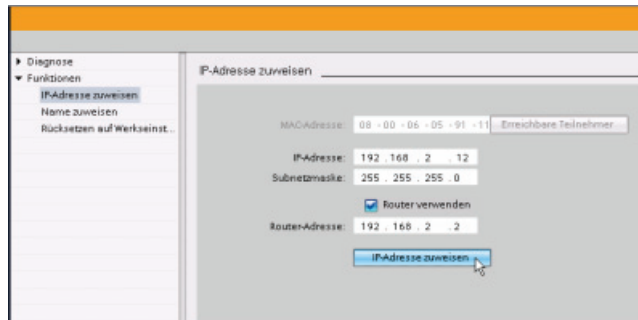
**HINWEIS:** Wenn statt einer IP-Adresse eine MAC-Adresse angezeigt wird, wurde keine IP-Adresse zugewiesen.

3. Wählen Sie im Dialog "Online & Diagnose" die folgenden Menübefehle:

- "Funktionen"
- "IP-Adresse zuordnen"

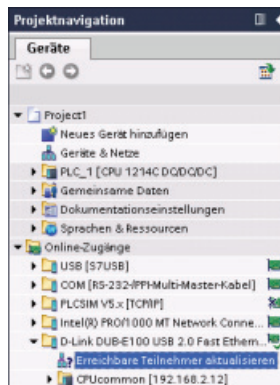


4. Geben Sie im Feld "IP-Adresse" Ihre neue IP-Adresse ein und klicken Sie auf die Schaltfläche "IP-Adresse zuweisen".



5. Prüfen Sie in der "Projektnavigation" mit den folgenden Befehlen, dass Ihre neue IP-Adresse der CPU zugewiesen wurde:

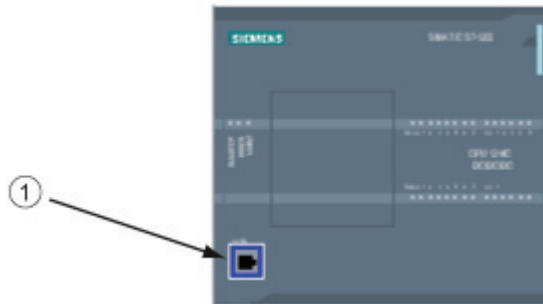
- "Online-Zugänge"
- <Adapterkarte für das Netzwerk, in dem sich das Gerät befindet>
- "Erreichbare Teilnehmer aktualisieren"



#### 5.6.4.4 IP-Adresse für eine CPU in Ihrem Projekt konfigurieren

##### Konfigurieren der PROFINET-Schnittstelle

Um Parameter für die PROFINET-Schnittstelle zu konfigurieren, wählen Sie das grüne PROFINET-Feld auf der CPU. Im Inspektorfenster wird das Register "Eigenschaften" für den PROFINET-Port angezeigt.



① PROFINET-Port

##### Konfigurieren der IP-Adresse

**Ethernet-Adresse (MAC-Adresse):** In einem PROFINET-Netzwerk ist jedem Gerät für die Identifikation eine MAC-Adresse (Media-Access-Control-Adresse) vom Hersteller zugewiesen. Eine MAC-Adresse besteht aus sechs Gruppen zu je zwei Hexadezimalziffern mit Bindestrichen (-) oder Doppelpunkten (:) als Trennzeichen in der Reihenfolge der Übertragung (zum Beispiel 01-23-45-67-89-AB oder 01:23:45:67:89:AB).

**IP-Adresse:** Jedes Gerät benötigt außerdem eine IP-Adresse (Internet-Protocol-Adresse). Mit Hilfe dieser Adresse kann ein Gerät Daten über ein komplexeres Netzwerk liefern.

Jede IP-Adresse ist in vier 8-Bit-Segmente unterteilt und wird in einem Dezimalformat mit Punkttrennzeichen dargestellt (zum Beispiel 211.154.184.16). Der erste Teil der IP-Adresse ist die Netzwerk-ID (Kennung des aktiven Netzwerks), der zweite Teil der Adresse ist die Host-ID (für jedes Gerät im Netzwerk eindeutig). Eine IP-Adresse von 192.168.x.y ist eine Standardkennzeichnung, die als Teil eines privaten Netzwerks, das sich nicht im Internet befindet, erkannt wird.

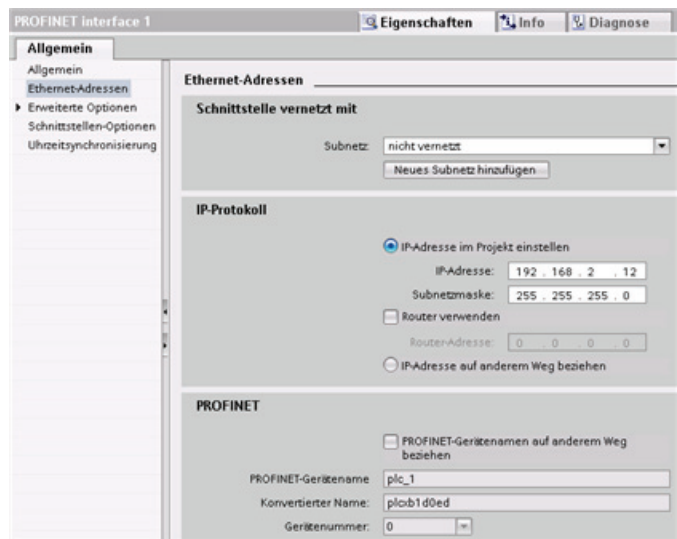
**Subnetzmaske:** Ein Subnetz ist eine logische Gruppierung miteinander verbundener Netzwerkgeräte. Die Teilnehmer eines Subnetzes befinden sich meist dicht nebeneinander auf einem LAN (Local Area Network). Eine Maske (die Subnetz- oder Netzmaske) legt die Grenzen eines IP-Subnetzes fest.

Eine Subnetzmaske von 255.255.255.0 eignet sich allgemein für ein kleines lokales Netzwerk. Das bedeutet, dass alle IP-Adressen in diesem Netzwerk dieselben ersten 3 Oktette haben müssen. Die einzelnen Geräte in diesem Netzwerk werden dann über das letzte Oktett (8-Bit-Feld) identifiziert. Ein Beispiel hierfür ist die Zuweisung einer Subnetzmaske von 255.255.255.0 und IP-Adressen von 192.168.2.0 bis 192.168.2.255 zu den einzelnen Geräten in einem kleinen lokalen Netzwerk.

Die einzige Verbindung zwischen verschiedenen Subnetzen läuft über einen Router. Werden Subnetze verwendet, so muss ein IP-Router eingesetzt werden.

**IP-Router:** Router sind die Verbindung zwischen LANs. Mit einem Router kann ein Rechner in einem LAN Meldungen an andere Netzwerke senden, die wiederum zu anderen LANs gehören. Liegt das Ziel der Daten nicht innerhalb des LANs, so leitet der Router die Daten an ein anderes Netzwerk oder eine Gruppen von Netzwerken weiter, wo die Daten ihrem Ziel zugestellt werden können.

Router benötigen IP-Adressen zum Senden und Empfangen von Datenpaketen.



**Eigenschaften von IP-Adressen:** Wählen Sie im Dialog "Eigenschaften" den Eintrag "Ethernet-Adressen". STEP 7 zeigt den Dialog für die Konfiguration der Ethernet-Adresse an, in dem Sie dem Softwareprojekt die IP-Adresse der CPU zuweisen, in die das Projekt geladen wird.

Tabelle 5-9 Parameter für die IP-Adresse

Parameter	Beschreibung	
Subnetz	Name des Subnetzes, mit dem das Gerät verbunden ist. Um ein neues Subnetz anzulegen, klicken Sie auf Schaltfläche "Neues Subnetz hinzufügen". Die Voreinstellung ist "nicht verbunden". Zwei Arten von Verbindungen sind möglich: <ul style="list-style-type: none"> <li>Die Voreinstellung "nicht verbunden" stellt eine lokale Verbindung her.</li> <li>Wenn Ihr Netzwerk über zwei oder mehr Geräte verfügt, ist ein Subnetz erforderlich.</li> </ul>	
IP-Protokoll	IP-Adresse	Zugewiesene IP-Adresse der CPU
	Subnetzmaske	Zugewiesene Subnetzmaske
	IP-Router verwenden	Aktivieren Sie das Optionskästchen, wenn ein IP-Router verwendet wird
	Router-Adresse	Zugewiesene IP-Adresse des Routers, sofern vorhanden

**Hinweis**

Alle IP-Adressen werden beim Laden des Projekts konfiguriert. Wenn die CPU keine vorkonfigurierte IP-Adresse besitzt, müssen Sie das Projekt mit der MAC-Adresse des Zielgeräts zuweisen. Ist Ihre CPU an einen Router oder ein Netzwerk angeschlossen, so muss auch die IP-Adresse des Routers eingegeben werden.

Über das Optionsfeld "IP-Adresse auf anderem Weg einstellen" können Sie die IP-Adresse online oder über die Anweisung "T\_CONFIG" ändern, nachdem Sie das Programm in die CPU geladen haben. Diese Zuweisungsart der IP-Adresse gilt nur für die CPU.

 **WARNUNG**

Nach dem Laden einer Hardwarekonfiguration mit aktivierter Option "IP-Adresse auf anderem Weg einstellen" ist es nicht möglich, den Betriebszustand der CPU von RUN in STOP oder von STOP in RUN zu versetzen.

Die Anwendergeräte laufen unter diesen Bedingungen weiter und es kann zu unvorhersehbarem Betrieb der Maschine bzw. des Prozesses kommen, was tödliche oder schwere Verletzungen oder Sachschaden verursachen kann, wenn keine geeigneten Sicherheitsvorkehrungen vorgenommen werden.

Stellen Sie sicher, dass die IP-Adresse(n) Ihrer CPU festgelegt sind, bevor Sie die CPU in einer echten Automatisierungsumgebung einsetzen. Für diese Überprüfung können Sie das Programmierpaket STEP 7, das S7-1200 Tool oder ein angeschlossenes HMI-Gerät in Verbindung mit der Anweisung T\_CONFIG nutzen.

 **WARNUNG**

Wenn Sie die IP-Adresse einer CPU online oder über das Anwenderprogramm ändern, kann dies dazu führen, dass das PROFINET-Netzwerk stoppt.

Wird die IP-Adresse einer CPU durch eine IP-Adresse außerhalb des Subnetzes ersetzt, verliert das PROFINET-Netzwerk die Kommunikation und der gesamte Datenaustausch stoppt. Die Anwendergeräte können so konfiguriert werden, dass sie unter diesen Bedingungen weiterarbeiten. Der Verlust der PROFINET-Kommunikation kann zu unvorhersehbarem Betrieb der Maschine bzw. des Prozesses führen und tödliche oder schwere Verletzungen oder Sachschaden verursachen, wenn keine geeigneten Sicherheitsvorkehrungen vorgenommen werden.

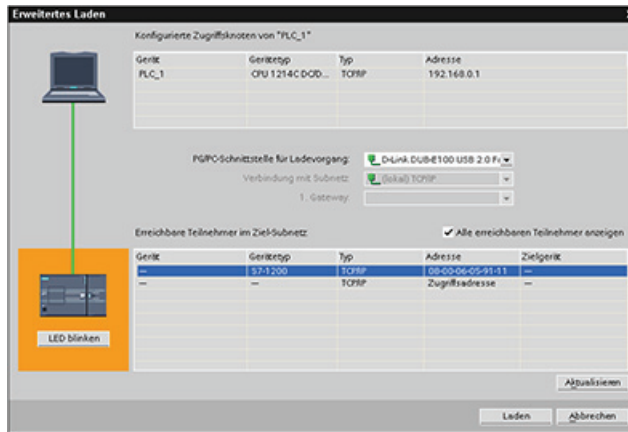
Wenn eine IP-Adresse manuell geändert werden muss, stellen Sie sicher, dass sich die neue IP-Adresse innerhalb des Subnetzes befindet.

**Siehe auch**

T\_CONFIG (Seite 479)

### 5.6.5 Testen des PROFINET-Netzwerks

Nachdem die Konfiguration beendet ist, laden Sie das Programm (Seite 179) in die CPU. Alle IP-Adressen werden beim Laden des Projekts konfiguriert.



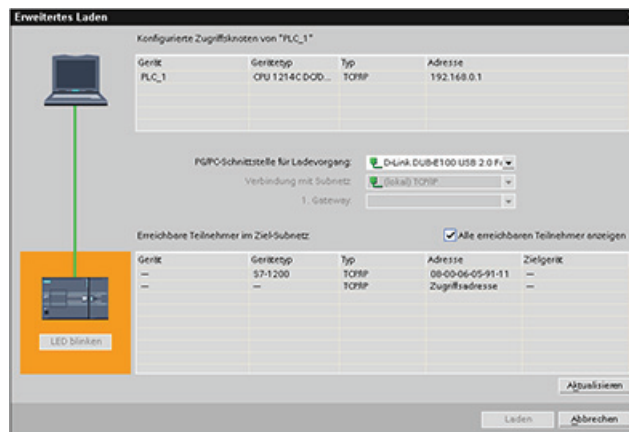
### Online-Zuweisung einer IP-Adresse zu einem Gerät

Die S7-1200 CPU hat keine vorkonfigurierte IP-Adresse. Sie müssen der CPU daher manuell eine IP-Adresse zuweisen.

- Wie Sie einem Gerät online eine IP-Adresse zuweisen, erfahren Sie unter "Gerätekonfiguration: IP-Adresse einer CPU online zuweisen" (Seite 145).
- Für die Zuweisung einer IP-Adresse in Ihrem Projekt muss die IP-Adresse in der Gerätekonfiguration eingerichtet, die Konfiguration gespeichert und ins Zielsystem geladen werden. Weitere Informationen finden Sie unter "Gerätekonfiguration: IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 147).

### Im Dialog "Erweitertes Laden" auf angeschlossene Netzwerkgeräte abfragen

Die Funktion "Laden in Gerät" der S7-1200 CPU und der zugehörigen Dialog "Erweitertes Laden" kann alle erreichbaren Netzwerkgeräte anzeigen und zusätzlich angeben, ob allen Geräten eindeutige IP-Adressen zugeordnet wurden. Um alle erreichbaren und verfügbaren Geräte mit ihren zugeordneten MAC- oder IP-Adressen anzuzeigen, aktivieren Sie das Optionskästchen "Alle erreichbaren Teilnehmer anzeigen".



Ist das erforderliche Netzwerkgerät in dieser Liste nicht enthalten, so wurde die Kommunikation mit diesem Gerät aus irgendeinem Grund unterbrochen. Das Gerät und das Netzwerk sind dann auf Hardware- und/oder Konfigurationsfehler zu überprüfen.

### 5.6.6 Ermitteln der Ethernet-Adresse (MAC-Adresse) der CPU

In PROFINET-Netzwerken ist die MAC-Adresse (Media-Access-Control-Adresse) eine Kennung, die den Netzwerkschnittstellen zur eindeutigen Kennzeichnung vom Hersteller zugewiesen wird. Die MAC-Adresse enthält üblicherweise die registrierte Identifikationsnummer des Herstellers.

Das Standardformat (IEEE 802.3) für den Druck von MAC-Adressen als Klartext besteht aus sechs Gruppen zu je zwei Hexadezimalziffern mit Bindestrichen (-) oder Doppelpunkten (:) als Trennzeichen in der Reihenfolge der Übertragung (zum Beispiel 01-23-45-67-89-ab oder 01:23:45:67:89:ab).

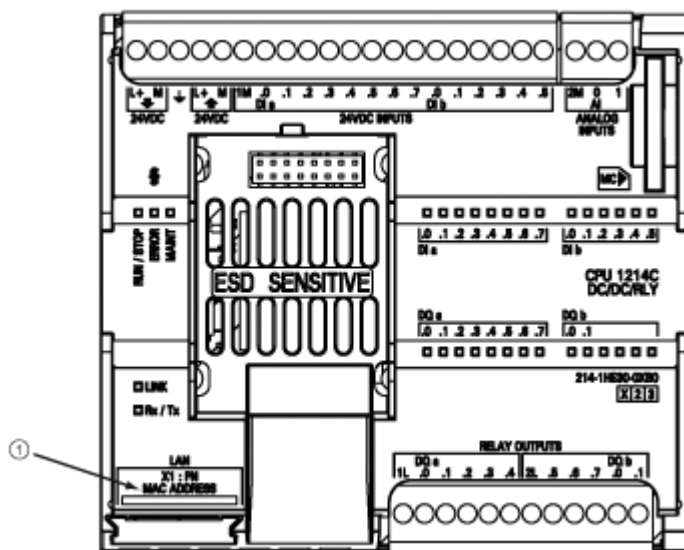
---

#### Hinweis

In jeder CPU wird im Werk eine dauerhafte, eindeutige MAC-Adresse eingestellt. Die MAC-Adresse einer CPU können Sie nicht ändern.

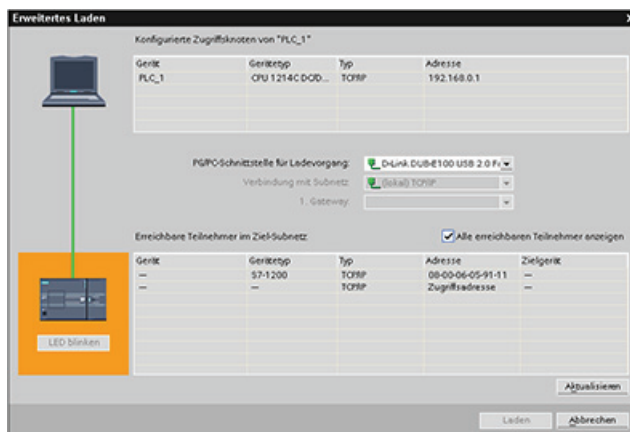
---

Die MAC-Adresse ist an der Vorderseite links unten auf der CPU aufgedruckt. Sie müssen die untere Abdeckklappe öffnen, um die MAC-Adresse lesen zu können.



① MAC-Adresse

Die CPU hat zunächst keine IP-Adresse, sondern nur eine im Werk eingestellte MAC-Adresse. In der PROFINET-Kommunikation ist es erforderlich, dass allen Geräten eine eindeutige IP-Adresse zugeordnet wird.



Über die Funktion "Laden in Gerät" der CPU und den Dialog "Erweitertes Laden" können Sie alle erreichbaren Teilnehmer im Netzwerk anzeigen und zusätzlich prüfen, ob allen Geräten eindeutige IP-Adressen zugeordnet wurden. Dieser Dialog zeigt alle erreichbaren und verfügbaren Geräte mit ihren zugeordneten MAC- oder IP-Adressen an. MAC-Adressen sind besonders wichtig, um Geräte zu identifizieren, die keine eindeutige IP-Adresse haben.

### 5.6.7 Konfigurieren der NTP-Synchronisierung

NTP (Network Time Protocol) ist weit verbreitet und dient der Synchronisierung der Uhren von Computersystemen mit Uhrzeitservern im Internet. Beim NTP-Verfahren sendet der CP in regelmäßigen Zeitabständen Uhrzeitanfragen (im Client-Modus) an den NTP-Server im Subnetz (LAN). Anhand der Antworten der Server wird die zuverlässigste und genaueste Uhrzeit ermittelt und die Uhrzeit der Station synchronisiert.

Der Vorteil dieses Verfahrens ist, dass es Zeit für die subnetzübergreifende Synchronisierung gestattet.



Es müssen die IP-Adressen von bis zu vier NTP-Servern konfiguriert werden. Das Aktualisierungsintervall legt den Zeitraum zwischen den Uhrzeitabfragen fest (in Sekunden). Der Wert des Intervalls liegt zwischen 10 Sekunden und einem Tag.

Beim NTP-Verfahren wird im Allgemeinen UTC (Universal Time Coordinated) übertragen, dies entspricht GMT (Greenwich Mean Time).

Wählen Sie im Eigenschaftfenster den Eintrag "Uhrzeitsynchronisation". In STEP 7 wird der Dialog "Uhrzeitsynchronisation" angezeigt:

### Hinweis

Alle IP-Adressen werden beim Laden des Projekts konfiguriert.

Tabelle 5- 10 Parameter für die Uhrzeitsynchronisation

Parameter	Definition
Uhrzeitsynchronisation mit NTP-Servern freigeben	Aktivieren Sie das Optionskästchen, um die Uhrzeitsynchronisation mit NTP-Servern freizugeben.
Server 1	Zugewiesene IP-Adresse für Netzwerkzeit-Server 1
Server 2	Zugewiesene IP-Adresse für Netzwerkzeit-Server 2
Server 3	Zugewiesene IP-Adresse für Netzwerkzeit-Server 3
Server 4	Zugewiesene IP-Adresse für Netzwerkzeit-Server 4
Intervall Uhrzeitsynchronisation	Intervallwert (s)

## 5.6.8 Anlaufzeit, Benennung und Adresszuweisung von PROFINET-Geräten

PROFINET IO kann die Anlaufzeit für Ihr System verlängern (konfigurierbarer Timeout-Wert). Ein größere Anzahl von Geräten und langsame Geräte wirken sich auf die Zeitdauer aus, die die Umschaltung in RUN benötigt.

Sie dürfen maximal die folgende Anzahl von PROFINET IO-Geräten in Ihrem S7-1200 PROFINET-Netzwerk haben:

- In der Version 3.0 dürfen maximal 16 IO-Geräte vorhanden sein.
- In der Version 2.2 dürfen maximal 8 IO-Geräte vorhanden sein.

Jede Station (bzw. IO-Gerät) startet beim Anlauf unabhängig und dies wirkt sich auf die Gesamtanlaufzeit der CPU aus. Wenn Sie den konfigurierbaren Timeout-Wert zu niedrig setzen, ist möglicherweise nicht genügend Gesamtanlaufzeit der CPU vorhanden, damit alle Stationen den Anlauf beenden. Tritt diese Situation ein, kommt es zu falschen Stationsfehlern.

Die "Zuweisungszeit der Parameter für dezentrale E/A" (Timeout) finden Sie in den CPU-Eigenschaften unter "Anlauf". Der Standardwert für den konfigurierbaren Timeout ist 60.000 ms (1 Minute). Diese Zeit lässt sich vom Anwender konfigurieren.

### Namen und Adressen von PROFINET-Geräten in STEP 7

Alle PROFINET-Geräte **müssen** einen Gerätenamen und eine IP-Adresse haben. Sie legen die Gerätenamen und die IP-Adressen in STEP 7 fest. Die Gerätenamen werden über PROFINET DCP (Discovery and Configuration Protocol, Erkennungs- und Konfigurationsprotokoll) in die IO-Geräte geladen.

### PROFINET-Adresszuweisung beim Anlauf des Systems

Die Steuerung überträgt die Namen der Geräte zum Netzwerk, und die Geräte reagieren mit ihren MAC-Adressen. Die Steuerung weist dem Gerät dann über das PROFINET DCP-Protokoll eine IP-Adresse zu:

- Wenn die MAC-Adresse eine konfigurierte IP-Adresse hat, führt die Station den Anlauf durch.
- Wenn die MAC-Adresse keine konfigurierte IP-Adresse hat, weist STEP 7 die im Projekt konfigurierte Adresse zu und die Station führt dann den Anlauf durch.
- Gibt es bei diesem Vorgang ein Problem, tritt ein Stationsfehler auf und es findet kein Anlauf statt. Diese Situation verursacht, dass der konfigurierbare Timeout-Wert überschritten wird.

## Programmiergrundlagen

### 6.1 Richtlinien für das Entwerfen einer Automatisierungslösung mit einem PLC-Gerät

Bei der Entwicklung eines PLC-Systems können Sie aus einer Vielzahl von Methoden und Kriterien auswählen. Die folgenden allgemeinen Richtlinien sind auf viele Projekte anwendbar. Dabei sollten Sie sich selbstverständlich an die Verfahrensanweisungen in Ihrem Unternehmen halten und Ihre eigenen Erfahrungen berücksichtigen.

Tabelle 6- 1 Richtlinien für das Entwerfen einer Automatisierungslösung mit einem PLC-Gerät

Empfohlene Schritte	Aufgaben
Gliedern Ihres Prozesses bzw. Ihrer Anlage	Unterteilen Sie Ihren Prozess bzw. Ihre Anlage in Abschnitte, die voneinander unabhängig sind. Diese Abschnitte legen die Grenzen zwischen mehreren Automatisierungssystemen fest und beeinflussen die Beschreibungen der Funktionsbereiche sowie die Zuordnung der Betriebsmittel.
Beschreiben der Funktionsbereiche	Geben Sie die Funktionsbeschreibungen für jeden Abschnitt des Prozesses bzw. der Anlage ein, wie z. B. Ein-/Ausgänge, die funktionale Beschreibung des Ablaufs, die Zustände, die erreicht werden müssen, bevor ein Aktor (z. B. ein Magnetventil, ein Motor oder ein Antrieb) reagieren darf, eine Beschreibung der Benutzeroberfläche und aller Schnittstellen zu anderen Abschnitten des Prozesses oder der Anlage.
Entwerfen der Sicherheitsstromkreise	Bestimmen Sie die Geräte, die aus Sicherheitsgründen festverdrahtete Schaltungen benötigen. Steuerungsgeräte können unsichere Betriebszustände einnehmen, woraus unerwartete Anlaufeigenschaften bzw. geänderte Funktionsabläufe der Anlage resultieren können. Besteht die Gefahr, dass bei unerwartetem bzw. fehlerhaftem Betrieb der Anlage schwere Körperverletzungen oder Sachschäden auftreten, sollten Sie die Einrichtung elektromechanischer Programmeingriffe in Erwägung ziehen, die unabhängig von der Steuerung arbeiten, um unsichere Betriebszustände zu vermeiden. Zum Entwerfen von Sicherheitsstromkreisen gehen Sie folgendermaßen vor: <ul style="list-style-type: none"> <li>• Definieren Sie falschen bzw. unerwarteten Betrieb von Aktoren, die Gefahrenpotentiale bergen.</li> <li>• Definieren Sie die Bedingungen, unter denen der Betrieb ungefährlich ist, und legen Sie fest, wie diese Bedingungen unabhängig von der SPS erkannt werden.</li> <li>• Definieren Sie, wie die SPS den Prozess beeinflusst, wenn die Spannung eingeschaltet und wieder ausgeschaltet wird und wenn Fehler erkannt werden. Verwenden Sie diese Informationen nur für den Entwurf des normalen und erwarteten unnormalen Betriebs. Verlassen Sie sich aus Sicherheitsgründen nicht auf dieses "Best-Case"-Szenario.</li> <li>• Entwerfen Sie Korrekturen durch Handeingriff bzw. elektromechanische Programmeingriffe, mit denen gefährliche Abläufe unabhängig von der SPS gesperrt werden.</li> <li>• Lassen Sie von den unabhängigen Stromkreisen Statusinformationen an die SPS übermitteln, so dass das Programm und jede Bedienerschnittstelle über die erforderlichen Informationen verfügt.</li> <li>• Definieren Sie weitere Sicherheitsanforderungen, damit der Prozess sicher ablaufen kann.</li> </ul>
Systemicherheit planen	Ermitteln Sie, welche Schutzstufe (Seite 176) Sie für den Zugriff auf Ihren Prozess benötigen. Sie können CPUs und Programmbausteine mittels Passwort vor unberechtigtem Zugriff schützen.

Empfohlene Schritte	Aufgaben
Definieren der OperatorStationen	Erstellen Sie die folgenden Pläne der OperatorStationen anhand der Anforderungen in den Beschreibungen der Funktionsbereiche: <ul style="list-style-type: none"> <li>• Position aller OperatorStationen in Bezug zum Prozess bzw. zur Anlage</li> <li>• Mechanische Anordnung der Geräte der Operator-Station, z. B. Display, Schalter und Lampen.</li> <li>• Verdrahtungspläne mit den zugehörigen Ein und Ausgängen der SPS und Signalmodule</li> </ul>
Zeichnen der Konfigurationspläne	Erstellen Sie die Konfigurationspläne für das Automatisierungssystem anhand der Anforderungen in den Beschreibungen der Funktionsbereiche: <ul style="list-style-type: none"> <li>• Positionen aller SPS in Bezug zum Prozess bzw. zur Anlage</li> <li>• Mechanische Anordnung aller SPS und E/A-Module (einschließlich Schränke usw.)</li> <li>• Verdrahtungspläne für alle SPS und E/A-Module, einschließlich Gerätenummern, Kommunikationsadressen und Adressen der Ein und Ausgänge</li> </ul>
Auflisten der symbolischen Namen	Erstellen Sie eine Liste der symbolischen Namen für die absoluten Adressen. Geben Sie nicht nur die physikalischen Ein und Ausgänge, sondern auch alle anderen Elemente (wie z. B. Variablennamen) an, die Sie in Ihrem Programm verwenden.

## 6.2 Strukturieren Ihres Anwenderprogramms

Beim Erstellen eines Anwenderprogramms für Automatisierungslösungen fügen Sie die Anweisungen für das Programm in Codebausteine ein:

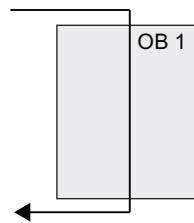
- Ein Organisationsbaustein (OB) reagiert auf ein bestimmtes Ereignis in der CPU und kann die Ausführung des Anwenderprogramms unterbrechen. Der Standardbaustein für die zyklische Ausführung des Anwenderprogramms (OB 1) stellt die Grundstruktur für Ihr Anwenderprogramm bereit und ist der einzige Codebaustein, der für das Anwenderprogramm erforderlich ist. Fügen Sie weitere OBs in Ihr Programm ein, so unterbrechen diese OBs die Ausführung von OB 1. Die anderen OBs führen spezifische Funktionen aus, wie z. B. für Anlauf, Interrupt- und Fehlerbearbeitung, oder für die Ausführung eines spezifischen Programmcodes in bestimmten Zeitabständen.
- Ein Funktionsbaustein (FB) ist ein Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Der aufrufende Baustein übergibt dem FB die Parameter und definiert einen bestimmten Datenbaustein (DB), der die Daten für diesen Aufruf oder diese Instanz des FBs speichert. Wird der Instanz-DB geändert, so kann ein allgemeiner FB den Betrieb einer Gerätegruppe steuern. Beispielsweise kann ein FB mehrere Pumpen oder Ventile mit unterschiedlichen Instanz-DBs, die die speziellen Betriebsparameter jeder Pumpe bzw. jedes Ventils enthalten, steuern.
- Eine Funktion (FC) ist eine Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Einer FC ist kein Instanz-DB zugeordnet. Der aufrufende Baustein übergibt der FC die Parameter. Die Ausgangswerte der FC müssen in eine Speicheradresse oder in einen globalen DB geschrieben werden.

## Art der Struktur für das Anwenderprogramm wählen

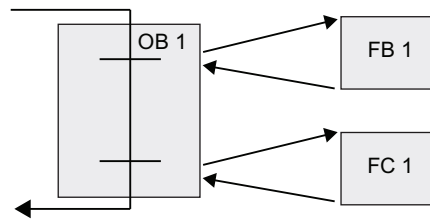
Je nach den Anforderungen Ihrer Anwendung können Sie eine lineare oder eine modulare Struktur für Ihr Anwenderprogramm wählen:

- Ein lineares Programm führt alle Anweisungen für Ihre Automatisierungsaufgaben nacheinander aus. Typischerweise werden bei einem linearen Programm alle Programmanweisungen im Zyklus-OB (OB 1) abgelegt.
- Ein modulares Programm ruft spezielle Codebausteine auf, die spezifische Aufgaben ausführen. Um eine modulare Programmstruktur aufzubauen, gliedern Sie die komplexe Automatisierungsaufgabe in kleinere Teilaufgaben, die den technologischen Funktionen des Prozesses entsprechen. Jeder Codebaustein enthält das Programmsegment für die jeweilige Teilaufgabe. Sie strukturieren Ihr Programm durch den Aufruf eines Codebausteins aus einem anderen Baustein.

Lineare Struktur:



Modulare Struktur:



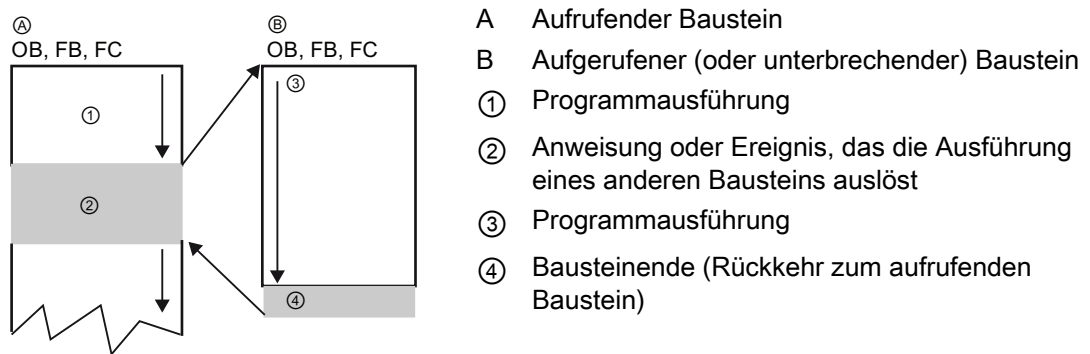
Durch das Anlegen allgemeiner Codebausteine, die im Anwenderprogramm mehrfach genutzt werden können, lassen sich Entwurf und Implementierung des Anwenderprogramms vereinfachen. Die Arbeit mit allgemeinen Codebausteinen hat mehrere Vorteile:

- Sie können wiederverwendbare Codebausteine für Standardaufgaben, wie z. B. für die Steuerung einer Pumpe oder eines Motors, erstellen. Sie können diese allgemeinen Codebausteine ferner in einer Bibliothek speichern, die für verschiedene Anwendungen oder Lösungen verwendet werden kann.
- Durch die modulare Gliederung des Anwenderprogramms in einzelne Komponenten, die sich auf funktionale Aufgaben beziehen, wird der Programmaufbau übersichtlicher und ist einfacher zu handhaben. Die modularen Komponenten ermöglichen nicht nur die Standardisierung des Programmentwurfs, sondern machen auch Anpassungen oder Änderungen des Programmcodes schneller und einfacher.
- Das Erstellen modularer Komponenten vereinfacht das Testen Ihres Programms. Wenn das komplette Programm in eine Folge modularer Programmsegmente gegliedert ist, können Sie die Funktionalität jedes Codebausteins direkt während der Entwicklung testen.
- Mit modularen Komponenten, die sich auf spezifische technologische Funktionen beziehen, wird die Inbetriebnahme der kompletten Anwendung vereinfacht und verkürzt.

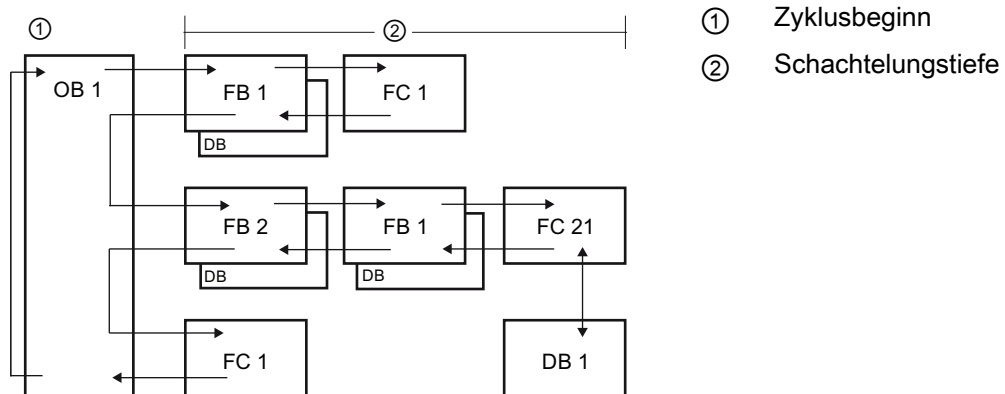
### 6.3 Strukturierung des Programms mit Hilfe von Bausteinen

Modulare Codebausteine erstellen Sie durch den Entwurf von FBs und FCs für die Ausführung allgemeiner Aufgaben. Dann strukturieren Sie Ihr Programm, indem andere Codebausteine diese wiederverwendbaren Module aufrufen. Der aufrufende Baustein gibt gerätespezifische Parameter an den aufgerufenen Baustein weiter.

Wird ein Codebaustein von einem anderen Codebaustein aufgerufen, führt die CPU den Programmcode im aufgerufenen Baustein aus. Nachdem der aufgerufene Baustein abgearbeitet ist, setzt die CPU die Ausführung des aufrufenden Bausteins fort. Die Bearbeitung geht weiter mit der Ausführung der nächsten Anweisung nach dem Bausteinaufruf.



Die Bausteinaufrufe können verschachtelt werden, um die Struktur noch modularer zu gestalten. Im folgenden Beispiel umfasst die Schachtelungstiefe 4 Ebenen: den Programmzyklus-OB plus 3 Ebenen mit Aufrufen von Codebausteinen.



### 6.3.1 Organisationsbaustein (OB)

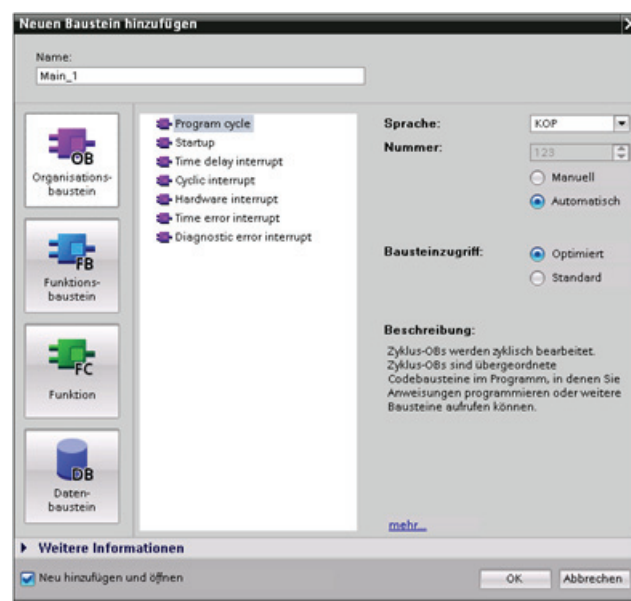
Organisationsbausteine dienen zur Strukturierung Ihres Programms. Sie bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. OBs sind ereignisgesteuert. Die Ausführung eines OBs durch das Zielsystem wird von einem Ereignis, wie z. B. einem Diagnosealarm oder einem Zeitintervall, angestoßen. Einige OBs haben voreingestellte Startereignisse und Startverhalten.

Der Programmzyklus-OB enthält das Hauptprogramm. Sie können mehrere Programmzyklus-OBs in Ihr Anwenderprogramm aufnehmen. Im Betriebszustand RUN werden die Programmzyklus-OBs mit der niedrigsten Prioritätsstufe ausgeführt und können durch alle anderen Arten der Programmverarbeitung unterbrochen werden. Der Anlauf-OB unterbricht den Programmzyklus-OB nicht, weil die CPU den Anlauf-OB vor dem Wechsel in RUN ausführt.

Nach Abarbeitung des Programmzyklus-OBs führt die CPU die Programmzyklus-OBs sofort erneut aus. Diese zyklische Ausführung ist die "normale" Ablaufart für speicherprogrammierbare Steuerungen. In vielen Anwendungen ist das gesamte Anwenderprogramm in einem einzigen Programmzyklus-OB enthalten.

Sie können andere OBs anlegen, um bestimmte Funktionen auszuführen, z. B. für die Alarm- und Fehlerbearbeitung oder für die Ausführung eines bestimmten Programmcodes in bestimmten Abständen. Diese OBs unterbrechen die Ausführung der Programmzyklus-OBs.

Neue OBs für Ihr Anwenderprogramm legen Sie im Dialog "Neuen Baustein hinzufügen" an.



Die Alarmbearbeitung ist immer ereignisgesteuert. Tritt ein solches Ereignis auf, so unterbricht die CPU die Ausführung des Anwenderprogramms und ruft den OB für die Behandlung dieses Ereignisses auf. Nach der Ausführung des unterbrechenden OBs setzt die CPU die Ausführung des Anwenderprogramms an der Stelle fort, an der es zuvor unterbrochen wurde.

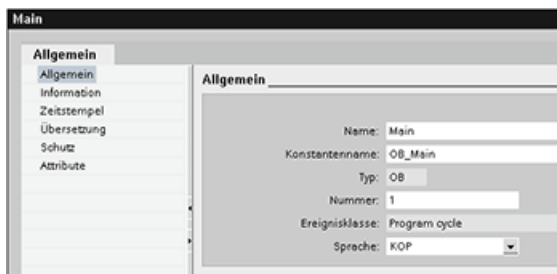
Die CPU ermittelt die Reihenfolge für die Bearbeitung von Alarmereignissen anhand der den einzelnen OBs zugeordneten Prioritäten. Jedes Ereignis hat eine bestimmte Bearbeitungspriorität. Die jeweilige Prioritätsstufe innerhalb einer Prioritätsklasse gibt die Reihenfolge vor, in der die OBs ausgeführt werden. Mehrere Alarmereignisse können in Prioritätsklassen zusammengeführt werden. Weitere Informationen finden Sie im Kapitel mit den PLC-Grundlagen im Abschnitt zur Ausführung des Anwenderprogramms (Seite 73).

### Anlegen eines weiteren OBs in einer OB-Klasse

Sie können mehrere OBs für Ihr Anwenderprogramm anlegen, auch für die Klassen der Programmzyklus- und Anlauf-OBs. Im Dialog "Neuen Baustein hinzufügen" können Sie einen OB anlegen. Geben Sie den Namen für Ihren OB und eine OB-Nummer ein, die größer als 200 sein muss.

Wenn Sie mehrere Programmzyklus-OBs für Ihr Anwenderprogramm anlegen, führt die CPU die einzelnen Programmzyklus-OBs in numerischer Reihenfolge aus. Dabei wird mit dem Programmzyklus-OB mit der niedrigsten Nummer begonnen (z. B. OB 1). Beispiel: Nach dem Ende des ersten Programmzyklus-OBs (z. B. OB 1) führt die CPU den nächsthöheren Programmzyklus-OB aus (z. B. OB 200).

### Konfigurieren der Funktionsweise eines OBs



Die Ablaufparameter eines OBs können Sie ändern. Beispielsweise können Sie die Zeit für einen Verzögerungs-OB oder für einen Zyklus-OB einstellen.

## 6.3.2 Funktion (FC)

Eine Funktion (FC) ist ein Codebaustein, der typischerweise eine bestimmte Operation mit einer Anzahl von Eingangswerten durchführt. Die FC speichert die Ergebnisse dieser Operation an bestimmten Speicherorten. Verwenden Sie beispielsweise FCs, um Standardoperationen und mehrfach verwendbare Operationen (z. B. mathematische Berechnungen) oder technologische Funktionen (z. B. für einzelne Steuerungsvorgänge über Bitverknüpfungsoperationen) durchzuführen. Eine FC kann auch mehrmals an verschiedenen Stellen eines Programms aufgerufen werden. Diese Wiederverwendung vereinfacht die Programmierung häufig wiederkehrender Aufgaben.

Einer FC ist kein zugehöriger Instanz-Datenbaustein(DB) zugeordnet. Die FC verwendet den lokalen Datenspeicher für die temporären Daten, die für die Berechnung der Operation benötigt werden. Die temporären Daten werden nicht gespeichert. Für die dauerhafte Datenspeicherung muss der Ausgangswert einem globalen Speicherplatz, wie z. B. dem Merkerspeicher, oder einem globalen DB zugewiesen werden.



### 6.3.3 Funktionsbaustein (FB)

Ein Funktionsbaustein (FB) ist ein Codebaustein, der für seine Parameter und statischen Daten einen Instanz-Datenbaustein nutzt. FBs haben einen variablen Speicher, der sich in einem Datenbaustein (DB) oder einem Instanz-DB befindet. Der Instanz-DB stellt einen Speicherbaustein bereit, der dieser Instanz (oder diesem Aufruf) des FBs zugewiesen ist und die Daten nach Ablauf des FBs speichert. Sie können verschiedenen Aufrufen des FBs verschiedene Instanz-DBs zuordnen. Der Instanz-DB ermöglicht es Ihnen, einen allgemeinen FB für die Steuerung mehrerer Geräte zu verwenden. Sie können Ihr Programm strukturieren, indem ein Codebaustein einen FB und einen Instanz-DB aufruft. Die CPU führt dann den Programmcode in diesem FB aus und speichert die Bausteinparameter und die statischen Lokaldaten im Instanz-DB. Wenn die Ausführung des FBs beendet ist, setzt die CPU die Ausführung mit dem Codebaustein fort, der den FB aufgerufen hatte. Der Instanz-DB speichert die Werte für diese Instanz des FBs. Diese Werte stehen nachfolgenden Aufrufen des Funktionsbausteins entweder in demselben Zyklus oder in anderen Zyklen zur Verfügung.

#### Wiederverwendbare Codebausteine mit zugewiesenem Speicher

Typischerweise wird ein FB für die Steuerung des Ablaufs von Tätigkeiten oder Geräten verwendet, deren Betrieb nicht innerhalb eines Zyklus endet. Für die Speicherung der Betriebsparameter, damit diese zwischen zwei Zyklen schnell zugänglich sind, hat jeder FB in Ihrem Anwenderprogramm einen oder mehrere Instanz-DBs. Mit dem Aufruf eines FBs geben Sie gleichzeitig einen Instanz-DB an, in dem die Bausteinparameter und die statischen Lokaldaten für diesen Aufruf oder diese "Instanz" des FBs enthalten sind. Der Instanz-DB speichert diese Werte nach der Ausführung des FBs.

Wird der FB für allgemeine Steuerungsaufgaben entworfen, so kann er für mehrere Geräte verwendet werden, indem verschiedene Instanz-DBs für die verschiedenen Aufrufe des FBs ausgewählt werden.

Ein FB speichert die Eingangs-, Ausgangs- und Durchgangsparameter sowie die statischen Parameter in einem Instanz-DB.

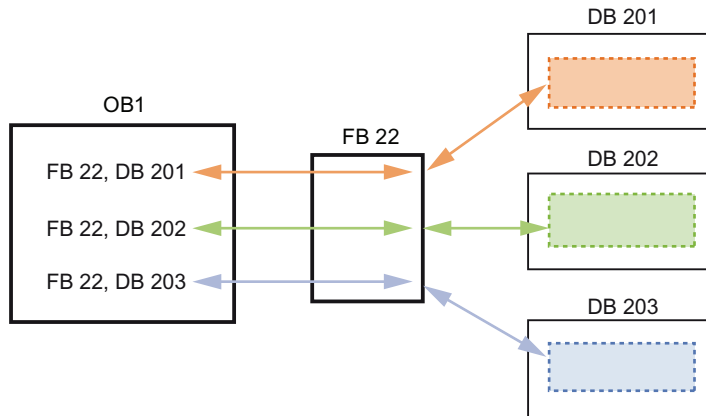
#### Startwert im Instanz-DB zuweisen

Der Instanz-DB speichert sowohl einen Standardwert als auch einen Startwert für jeden Parameter. Der Startwert gibt den Wert an, der verwendet werden soll, wenn der FB ausgeführt wird. Der Startwert kann dann während der Ausführung Ihres Anwenderprogramms geändert werden.

Die FB-Schnittstelle verfügt auch über eine Spalte für den Standardwert, in der Sie beim Schreiben des Programmcode einen neuen Startwert für den Parameter eingeben können. Dieser Standardwert im FB wird dann in den Startwert des zugehörigen Instanz-DBs übertragen. Wenn Sie in der FB-Schnittstelle einem Parameter keinen neuen Startwert zuweisen, wird der Standardwert aus dem Instanz-DB in den Startwert kopiert.

### Einzelnen FB mit DBs verwenden

Die folgende Abbildung zeigt einen OB, der einen FB dreimal aufruft, wobei für jeden Aufruf ein anderer Datenbaustein verwendet wird. Durch diese Struktur kann ein allgemeiner FB für die Steuerung mehrerer gleichartiger Geräte wie z. B. Motoren verwendet werden, indem jedem Aufruf eines Geräts ein anderer Instanzdatenbaustein zugewiesen wird. Jeder Instanz-DB speichert die Daten (wie Drehzahl, Hochlaufzeit und Gesamtbetriebszeit) für ein einziges Gerät.



In diesem Beispiel steuert FB 22 drei verschiedene Geräte, DB 201 speichert die Betriebsdaten für das erste Gerät, DB 202 die Betriebsdaten für das zweite Gerät und DB 203 die Betriebsdaten für das dritte Gerät.

### 6.3.4 Datenbaustein (DB)

Sie können in Ihrem Anwenderprogramm Datenbausteine (DBs) zum Speichern der Daten für die Codebausteine anlegen. Alle Programmbausteine im Anwenderprogramm können auf die Daten in einem globalen DB zugreifen, doch ein Instanz-DB speichert Daten für einen spezifischen Funktionsbaustein (FB).

Die gespeicherten Daten in einem DB werden nach der Ausführung des zugehörigen Codebausteins nicht gelöscht. Es gibt zwei Arten von Datenbausteinen:

- In einem globalen DB werden die Daten für die Codebausteine in Ihrem Programm gespeichert. Jeder OB, FB oder FC kann auf die Daten in einem globalen DB zugreifen.
- In einem Instanz-DB werden die Daten für einen spezifischen FB gespeichert. Die Datenstruktur in einem Instanz-DB entspricht den Parametern (Input, Output und InOut) und den statischen Daten des FBs. (Der temporäre Speicher des FBs wird im Instanz-DB nicht gespeichert.)

---

#### Hinweis

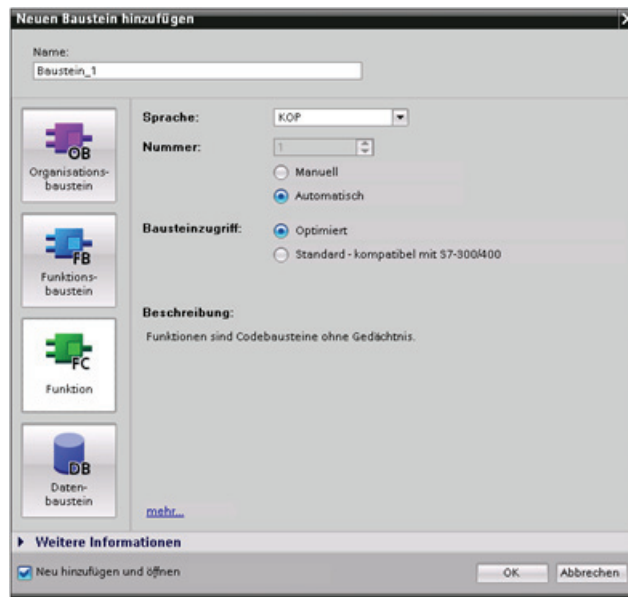
Obwohl der Instanz-DB die Daten für einen bestimmten FB enthält, kann jeder Codebaustein auf die Daten eines Instanz-DBs zugreifen.

---

Ein DB kann so konfiguriert werden, dass er nur gelesen werden kann:

1. Klicken Sie mit der rechten Maustaste in der Projektnavigation auf den DB und wählen Sie im Kontextmenü "Eigenschaften".
2. Wählen Sie Dialog "Eigenschaften" das Element "Attribute".
3. Wählen Sie die Option "Datenbaustein im Gerät schreibgeschützt" und bestätigen Sie mit "OK".

## Anlegen wiederverwendbarer Codebausteine



OBs, FBs, FCs und globale DBs können Sie in der Projektnavigation unter "Programmbausteine" im Dialog "Neuen Baustein hinzufügen" anlegen.

Beim Anlegen eines Codebausteins müssen Sie die Programmiersprache für den Baustein auswählen. Für einen DB wird keine Sprache ausgewählt, weil er nur Daten speichert.

## 6.4 Datenkonsistenz

Die CPU erhält die Datenkonsistenz für alle elementaren Datentypen (z.B. Word oder DWord) und alle systemdefinierten Strukturen (z.B. IEC\_TIMERS oder DTL). Der Lese- bzw. Schreibvorgang des Werts kann nicht unterbrochen werden. (Die CPU z.B. schützt den Zugriff auf einen Wert vom Datentyp DWord, bis die vier Bytes des DWord gelesen oder geschrieben wurden.) Um sicherzustellen, dass die Programmzyklus-OBs und die Alarm-OBs nicht gleichzeitig in dieselbe Adresse im Speicher schreiben können, führt die CPU einen Alarm-OB erst aus, wenn der Lese- oder Schreibvorgang im Programmzyklus-OB beendet ist.

Wenn in Ihrem Anwenderprogramm mehrere Werte im Speicher von einem Programmzyklus-OB und einem Alarm-OB gemeinsam genutzt werden, muss Ihr Anwenderprogramm auch sicherstellen, dass diese Werte konsistent geändert oder gelesen werden. Mit den Anweisungen DIS\_AIRT (Alarmbearbeitung deaktivieren) und EN\_AIRT (Alarmbearbeitung aktivieren) können Sie in Ihrem Programmzyklus-OB den Zugriff auf die gemeinsam genutzten Werte schützen.

- Fügen Sie eine Anweisung DIS\_AIRT in den Codebaustein ein, um sicherzustellen, dass während eines Lese- oder Schreibvorgangs kein Alarm-OB ausgeführt werden kann.
- Fügen Sie die Anweisungen ein, die die Werte, die von einem Alarm-OB geändert werden könnten, lesen oder schreiben.
- Fügen Sie am Ende der Sequenz eine Anweisung EN\_AIRT ein, um die Anweisung DIS\_AIRT zu stornieren und die Ausführung eines Alarm-OBs zu gestatten.

Auch eine Kommunikationsanforderung eines HMI-Geräts oder einer anderen CPU kann die Ausführung des Programmzyklus-OBs unterbrechen. Die Kommunikationsanforderungen können auch zu Datenkonsistenz führen. Die CPU stellt sicher, dass die elementaren Datentypen von den Anweisungen im Anwenderprogramm stets konsistent gelesen und geschrieben werden. Weil das Anwenderprogramm regelmäßig von Kommunikationsanforderungen unterbrochen wird, kann nicht gewährleistet werden, dass mehrere Werte der CPU alle gleichzeitig von der HMI aktualisiert werden. Die in einem HMI-Bild angezeigten Werte beispielsweise können aus unterschiedlichen Zyklen der CPU stammen.

Die PtP-Anweisungen (Punkt-zu-Punkt), PROFINET-Anweisungen (wie TSEND\_C und TRCV\_C), die PROFINET-Anweisungen für die dezentrale Peripherie und die PROFIBUS-Anweisungen für die dezentrale Peripherie (Seite 289) übertragen Datenpuffer, die unterbrochen werden können. Stellen Sie die Datenkonsistenz für die Datenpuffer dadurch sicher, dass Sie Lese- und Schreibvorgänge in den Puffern sowohl im Programmzyklus-OB als auch im Alarm-OB verhindern. Sollte es erforderlich sein, die Pufferwerte für diese Anweisungen in einem Alarm-OB zu ändern, verzögern Sie die Unterbrechung (durch einen Alarm-OB oder eine Kommunikationsunterbrechung von einem HMI-Gerät oder einer anderen CPU) mit der Anweisung DIS\_AIRT, bis eine Anweisung EN\_AIRT ausgeführt wird.

---

#### Hinweis

Die Verwendung der Anweisung DIS\_AIRT verzögert die Verarbeitung von Alarm-OBs, bis die Anweisung EN\_AIRT ausgeführt wird, und wirkt sich auf die Alarmlatenz (Zeitraum vom Auftreten eines Ereignisses bis zur Ausführung des Alarm-OBs) Ihres Anwenderprogramms aus.

---

## 6.5 Programmiersprache

STEP 7 bietet die folgenden Standardprogrammiersprachen für die S7-1200:

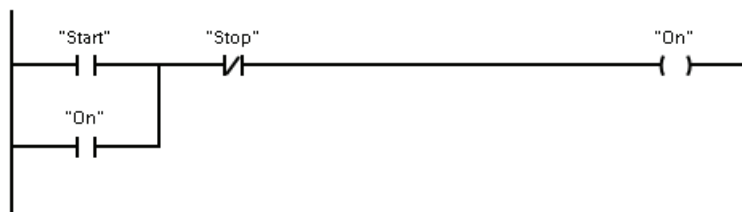
- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Darstellung beruht auf Schaltplänen (Seite 165).
- FUP (Funktionsplan) ist eine Programmiersprache, die auf den grafischen Logiksymbolen der Booleschen Algebra (Seite 166) basiert.
- SCL (Structured Control Language) ist eine textbasierte, höhere Programmiersprache (Seite 166).

Wenn Sie einen Codebaustein anlegen, müssen Sie die Programmiersprache für den Baustein auswählen.

Ihr Anwenderprogramm kann mit Codebausteinen arbeiten, die in einer dieser Programmiersprachen angelegt wurden.

### 6.5.1 Kontaktplan (KOP)

Die Elemente eines Schaltplans, wie Öffner- und Schließerkontakte, und Spulen werden zu Netzwerken verknüpft.



Um Verknüpfungen für komplexe Operationen anzulegen, können Sie Verzweigungen für parallele Kreise einfügen. Parallele Verzweigungen sind nach unten geöffnet oder direkt mit der Stromschiene verbunden. Sie beenden die Verzweigungen nach unten.

KOP bietet Box-Anweisungen für eine Vielzahl von Funktionen wie Arithmetik, Zeiten, Zähler und Übertragen.

STEP 7 begrenzt die maximale Anzahl von Anweisungen (Zeilen und Spalten) in einem KOP-Netzwerk nicht.

---

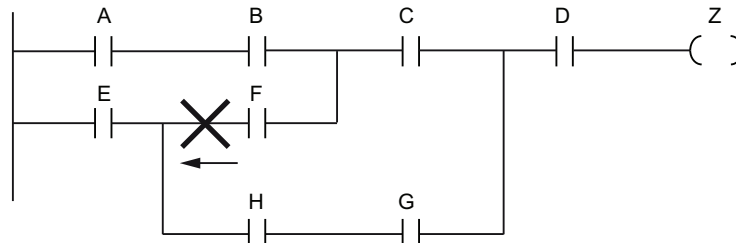
#### Hinweis

Jedes KOP-Netzwerk muss mit einer Spule oder einer Box abgeschlossen werden.

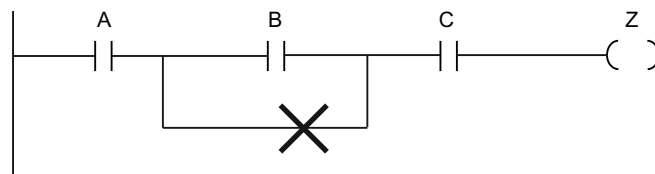
---

Beim Anlegen eines KOP-Netzwerks sind die folgenden Regeln zu beachten:

- Sie können keine Verzweigung anlegen, die zu einem Signalfluss in die Gegenrichtung führen könnte.

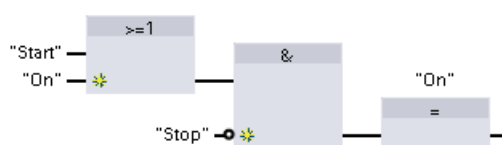


- Sie können keine Verzweigung anlegen, die einen Kurzschluss verursachen würde.



### 6.5.2 Funktionsplan (FUP)

Ebenso wie KOP ist auch FUP eine grafische Programmiersprache. Die Darstellung der Verknüpfungslogik beruht auf den grafischen Symbolen, die in der booleschen Algebra üblich sind.



Um Verknüpfungen für komplexe Operationen anzulegen, fügen Sie parallele Verzweigungen zwischen den Boxen ein.

Arithmetische Funktionen und andere komplexe Funktionen können direkt in Verbindung mit den Logikboxen dargestellt werden.

STEP 7 begrenzt die maximale Anzahl von Anweisungen (Zeilen und Spalten) in einem FUP-Netzwerk nicht.

### 6.5.3 SCL

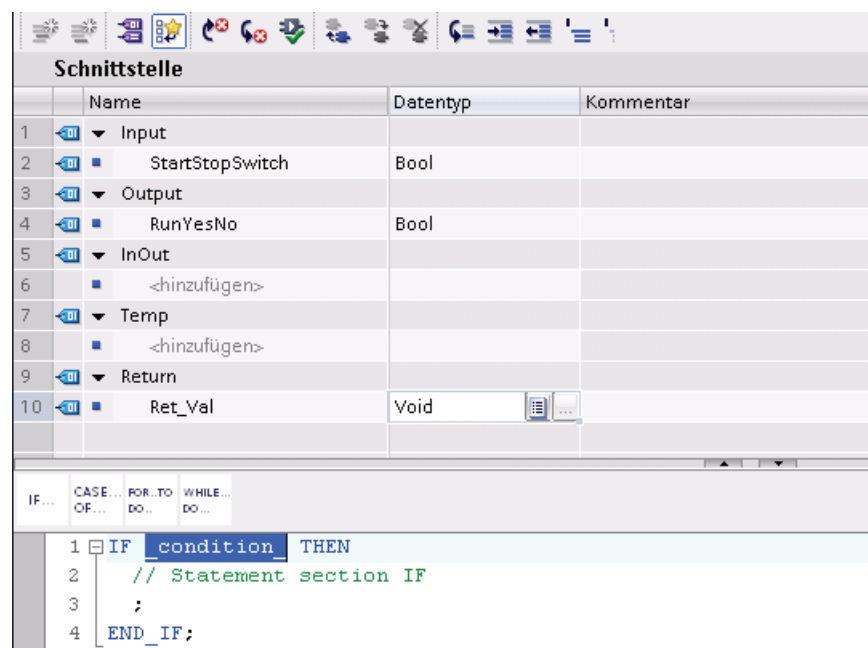
Structured Control Language (SCL) ist eine höhere, PASCAL-basierte Programmiersprache für die SIMATIC S7-CPU. SCL unterstützt die Bauelementstruktur von STEP 7 (Seite 158). Sie können in SCL geschriebene Programmbausteine mit in KOP und FUP geschriebenen Programmbausteinen kombinieren.

In SCL-Anweisungen werden die Standardoperatoren der Programmierung verwendet, z. B. für Zuweisung (:=), mathematische Funktionen (+ für Addition, - für Subtraktion, \* für Multiplikation und / für Division). SCL verwendet auch standardmäßige PASCAL-Operationen für die Programmsteuerung, z. B. IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO und RETURN. Für syntaktische Elemente der Programmiersprache SCL können Sie alle PASCAL-Referenzen verwenden. Viele der anderen Anweisungen für SCL wie Zeiten und Zähler entsprechen den Anweisungen in KOP und FUP. Weitere Informationen zu spezifischen Anweisungen finden Sie unter den jeweiligen Anweisungen in den Kapiteln Grundlegende Anweisungen (Seite 185) und Erweiterte Anweisungen (Seite 261).

Sie können für alle Bausteintypen (OB, FB oder FC) beim Erstellen des Bausteins angeben, dass er die Programmiersprache SCL verwenden soll. STEP 7 verfügt über einen SCL-Programmiereditor, der die folgenden Elemente enthält:

- Schnittstellenabschnitt zum Definieren der Parameter des Codebausteins
- Codeabschnitt für den Programmcode
- Anweisungsverzeichnis mit den SCL-Anweisungen, die von der CPU unterstützt werden

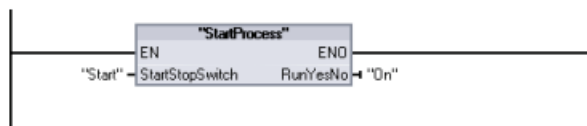
Sie geben den SCL-Code für Ihre Anweisung direkt in den Codeabschnitt ein. Um komplexere Anweisungen anzulegen, ziehen Sie die SCL-Anweisungen einfach mit der Maus aus dem Anweisungsverzeichnis in Ihr Programm. Ferner können Sie in jedem Texteditor ein SCL-Programm anlegen und die Datei dann in STEP 7 importieren.



Im Abschnitt mit dem SCL-Codebaustein können Sie die folgenden Arten von Parametern deklarieren:

- Eingang, Ausgang, Durchgang und Rückgabewert: Diese Parameter definieren die Eingangs- und Ausgangsvariablen sowie den Rückgabewert für den Codebaustein. Der Variablenname, den Sie hier eingeben, wird lokal während der Ausführung des Codebausteins verwendet. Üblicherweise wird der globale Variablenname nicht in der Variablen-tabelle verwendet.
- Statisch (nur FBs; die obige Abbildung zeigt einen FC): Statische Variablen dienen zum Speichern von statischen Zwischenergebnissen im Instanzdatenbaustein. Statische Daten werden gespeichert, bis sie überschrieben werden. d. h. zum Teil über mehrere Zyklen. Zusammen mit den statischen Daten werden auch die Namen der Bausteine gespeichert, die in diesem Codebaustein als Multiinstanz-DBs aufgerufen werden.
- Temp: Bei diesen Parametern handelt es sich um temporäre Variablen, die während der Ausführung des Codebausteins verwendet werden.

Wenn Sie den SCL-Codebaustein aus einem anderen Codebaustein aufrufen, treten die Parameter des SCL-Codebausteins als Eingänge oder Ausgänge auf.



In diesem Beispiel entsprechen die Variablen für "Start" und "On" (aus der Variablen-tabelle des Systems) den Variablen "StartStopSwitch" und "RunYesNo" in der Deklarationstabelle des SCL-Programms.

## SCL-Ausdruck konstruieren

Ein SCL-Ausdruck ist eine Formel zum Berechnen eines Werts. Der Ausdruck besteht aus Operanden und Operatoren (wie \*, /, + oder -). Bei den Operanden kann es sich um Variablen, Konstanten oder Ausdrücke handeln.

Die Auswertung des Ausdrucks erfolgt in einer bestimmten Reihenfolge, die von den folgenden Faktoren festgelegt wird:

- Jeder Operator hat eine vordefinierte Priorität, wobei die Operation mit der höchsten Priorität zuerst ausgeführt wird.
- Bei Operatoren mit gleicher Priorität werden die Operatoren von links nach rechts verarbeitet.
- Mit Hilfe von Klammern kennzeichnen Sie eine Reihe von Operatoren, die gemeinsam ausgewertet werden sollen.

Mit dem Ergebnis eines Ausdrucks kann ein Wert einer von Ihrem Programm verwendeten Variablen zugewiesen werden, das Ergebnis kann als Bedingung für eine Steuerungsanweisung oder als Parameter für eine andere SCL-Anweisung oder zum Aufrufen eines Codebausteins verwendet werden.



Tabelle 6- 2 Operatoren in SCL

Typ	Operation	Operator	Priorität
Klammern	(Ausdruck)	( , )	1
Arithmetik	Potenz	**	2
	Vorzeichen (unäres Plus)	+	3
	Vorzeichen (unäres Minus)	-	3
	Multiplikation	*	4
	Division	/	4
	Modulo-Funktion	MOD	4
	Addition	+	5
	Subtraktion	-	5
Vergleich	Kleiner als	<	6
	Kleiner oder gleich	<=	6
	Größer als	>	6
	Größer oder gleich	>=	6
	Gleich	=	7
	Ungleich	<>	7
Bitverknüpfung	Negation (unär)	NOT	3
	Logische UND-Verknüpfung	AND oder &	8
	Logische Exklusiv-ODER-Verknüpfung	XOR	9
	Logische ODER-Verknüpfung	OR	10
Zuweisung	Zuweisung	:=	11

Als höhere Programmiersprache nutzt SCL Standardanweisungen für grundlegende Aufgaben:

- Zuweisungsanweisung: :=
- Mathematische Funktionen: +, -, \* und /
- Adressierung von globalen Variablen: "<Variablenname>" (Variablenname oder Datenbausteinname in doppelten Anführungszeichen)
- Adressierung von lokalen Variablen: #<Variablenname> (Variablenname mit vorangestelltem Symbol "#")

Die folgenden Beispiele zeigen verschiedene Ausdrücke für verschiedene Einsatzzwecke.

"C" := #A+#B;	Weist einer Variablen die Summe zweier lokaler Variablen zu
"Data_block_1".Tag := #A;	Zuweisung zu einer Datenbausteinvariablen
IF #A > #B THEN "C" := #A;	Bedingung für die IF-THEN-Anweisung
"C" := SQRT (SQR (#A) + SQR (#B));	Parameter für die SQRT-Anweisung

Arithmetische Operatoren können verschiedene numerische Datentypen verarbeiten. Der Datentyp des Ergebnisses wird vom Datentyp des höchstwertigen Operanden festgelegt. Beispiel: Eine Multiplikationsoperation mit einem Operanden vom Typ INT und einem Operanden vom Typ REAL ergibt einen Wert vom Typ REAL als Ergebnis.

## Steuerungsanweisungen

Eine Steuerungsanweisung ist eine besondere Art von SCL-Ausdruck, der die folgenden Aufgaben durchführt:

- Programmverzweigung
- Wiederholung von Abschnitten des SCL-Programmcodes
- Sprung zu anderen Teilen des SCL-Programms
- Bedingte Ausführung

Die SCL-Steuerungsanweisungen umfassen IF-THEN, CASE-OF, FOR-TO-DO, WHILE-DO, REPEAT-UNTIL, CONTINUE, GOTO und RETURN.

Eine einzelne Anweisung belegt üblicherweise eine Codezeile. Sie können mehrere Anweisungen in einer Zeile eingeben oder Sie können die Anweisung auf mehrere Codezeilen verteilen, damit der Code besser lesbar ist. Trennzeichen (wie Tabulatoren, Zeilenumbrüche und zusätzliche Leerzeichen) werden während der Syntaxprüfung ignoriert. Die END-Anweisung beendet die Steuerungsanweisung.

Die folgenden Beispiele zeigen eine FOR-TO-DO-Steuerungsanweisung. (Beide Formen der Codierung sind syntaktisch gültig.)

```
FOR x := 0 TO max DO sum := sum + value(x); END_FOR;  
FOR x := 0 TO max DO  
    sum := sum + value(x);  
END_FOR;
```

Eine Steuerungsanweisung kann auch eine Sprungmarke umfassen. Eine Sprungmarke wird gefolgt von einem Doppelpunkt an den Anfang der Anweisung gesetzt:

**Sprungmarke:** <Anweisung>;

Die Online-Hilfe von STEP 7 enthält vollständige Referenzinformationen für die Programmiersprache SCL.

## Bedingungen

Eine Bedingung ist ein Vergleichsausdruck oder ein logischer Ausdruck, dessen Ergebnis vom Typ BOOL ist (Wert WAHR oder FALSCH). Die folgenden Beispiele zeigen Bedingungen verschiedener Arten.

#Temperatur > 50	Relationaler Ausdruck
#Zähler <= 100	
#CHAR1 < 'S'	
(#Alpha <> 12) AND NOT #Beta	Vergleichsausdruck und logischer Ausdruck
5 + #Alpha	Arithmetischer Ausdruck

Eine Bedingung kann arithmetische Ausdrücke verwenden:

- Die Bedingung des Ausdrucks ist WAHR, wenn das Ergebnis ein beliebiger Wert außer Null ist.
- Die Bedingung des Ausdrucks ist FALSCH; wenn das Ergebnis null ist.

## Adressierung

Wie bei KOP und FUP können Sie in SCL entweder Variablen (symbolische Adressierung) oder absolute Adressen in Ihrem Anwenderprogramm verwenden. In SCL können Sie eine Variable auch als Array-Index verwenden.

### Absolute Adressierung

E0.0  
MB100

### Symbolische Adressierung

"PLC_Tag_1"	Variable in PLC-Variablen-tabelle
"Data_block_1".Tag_1	Variable in einem Datenbaustein
"Data_block_1".MyArray[#i]	Array-Element in einem Datenbaustein-Array

## Indexierte Adressierung mit den Anweisungen PEEK und POKE

SCL bietet die Anweisungen PEEK und POKE, mit denen Sie aus Datenblöcken, E/A oder dem Speicher lesen oder in diese schreiben können. Sie geben für die Operation Parameter mit einem spezifischen Byte- oder Bit-Versatz an.

---

### Hinweis

Um die Anweisungen PEEK und POKE mit Datenbausteinen zu verwenden, müssen Sie Standarddatenbausteine (keine optimierten Datenbausteine) verwenden. Beachten Sie zudem, dass die Anweisungen PEEK und POKE lediglich Daten übertragen. Die Datentypen an den Adressen werden nicht beachtet.

---

```
PEEK(area:=_in_,  
      dbNumber:=_in_,  
      byteOffset:=_in_);
```

Liest das von byteOffset angegebene Byte aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel für den Verweis auf einen Datenbaustein:

```
%MB100 := PEEK(area:=16#84,  
              dbNumber:=1, byteOffset:=#i);
```

Beispiel für den Verweis auf Eingang EB3:

```
%MB100 := PEEK(area:=16#81,  
              dbNumber:=0, byteOffset:=#i); // when  
#i = 3
```

```
PEEK_WORD(area:=_in_,  
          dbNumber:=_in_,  
          byteOffset:=_in_);
```

Liest das von byteOffset angegebene Wort aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MW200 := PEEK_WORD(area:=16#84,  
                   dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,  
            dbNumber:=_in_,  
            byteOffset:=_in_);
```

```
PEEK_BOOL(area:=_in_,  
           dbNumber:=_in_,  
           byteOffset:=_in_,  
           bitOffset:=_in_);
```

```
POKE(area:=_in_,  
      dbNumber:=_in_,  
      byteOffset:=_in_,  
      value:=_in_);
```

Liest das von byteOffset angegebene Doppelwort aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MD300 := PEEK_DWORD(area:=16#84,  
                    dbNumber:=1, byteOffset:=#i);
```

Liest einen von bitOffset und byteOffset angegebenen Booleschen Wert aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MB100.0 := PEEK_BOOL(area:=16#84,  
                     dbNumber:=1, byteOffset:=#ii,  
                     bitOffset:=#j);
```

Schreibt den Wert (Byte, Wort oder Doppelwort) in den angegebenen byteOffset des angegebenen Datenbausteins, der E/A oder des Speicherbereichs.

Beispiel für den Verweis auf einen Datenbaustein:

```
POKE(area:=16#84, dbNumber:=2,  
      byteOffset:=3, value:="Tag_1");
```

Beispiel für den Verweis auf Ausgang AB3:

```
POKE(area:=16#82, dbNumber:=0,  
      byteOffset:=3, value:="Tag_1");
```

```
POKE_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_,
           value:=_in_);
```

Schreibt den Booleschen Wert in den angegebenen bitOffset und byteOffset des angegebenen Datenbausteins, der E/A oder des Speicherbereichs.

Beispiel:

```
POKE_BOOL(area:=16#84, dbNumber:=2,
           byteOffset:=3, bitOffset:=5,
           value:=0);
```

```
POKE_BLK(area_src:=_in_,
          dbNumber_src:=_in_,
          byteOffset_src:=_in_,
          area_dest:=_in_,
          dbNumber_dest:=_in_,
          byteOffset_dest:=_in_,
          count:=_in_);
```

Schreibt die unter "count" angegebene Anzahl von Bytes beginnend mit dem angegebenen byteOffset des angegebenen Quelldatenbausteins, der Quell-E/A oder des Quell Speicherbereichs in den angegebenen byteOffset des angegebenen Zieldatenbausteins, der Ziel-E/A oder des Zielspeicherbereichs.

Beispiel:

```
POKE_BLK(area_src:=16#84,
          dbNumber_src:=#src_db,
          byteOffset_src:=#src_byte,
          area_dest:=16#84,
          dbNumber_dest:=#src_db,
          byteOffset_dest:=#src_byte,
          count:=10);
```

Bei den Anweisungen PEEK und POKE gelten die folgenden Werte für die Parameter "area", "area\_src" und "area\_dest". Für andere Bereiche als Datenbausteine muss der Parameter dbNumber 0 sein.

16#81	I
16#82	Q
16#83	M
16#84	DB

## Andere Codebausteine aus Ihrem SCL-Programm aufrufen

Um einen anderen Codebaustein in Ihrem Anwenderprogramm aufzurufen, geben Sie einfach den Namen (oder die absolute Adresse) von FB oder FC mit den Parametern ein. Für einen FB müssen Sie den Instanz-DB angeben, der mit dem FB aufgerufen werden soll.

<DB-Name> (Parameterliste)

Aufruf als eine Instanz

<#Instanzname> (Parameterliste)

Aufruf als Multiinstanz

```
"MyDB" (MyInput:=10, MyInOut:="Tag1");
```

<FC-Name> (Parameterliste)	Standardaufruf
<Operand>:=<FC-Name> (Parameterliste)	Aufruf in einem Ausdruck

```
"MyFC" (MyInput:=10, MyInOut:="Tag1") ;
```

Sie können Bausteine auch mit der Maus aus der Projektnavigation in den SCL-Programmieditor ziehen und die Parametrierung vervollständigen.

### 6.5.4 EN und ENO in KOP, FUP und SCL

#### "Signalfluss" (EN und ENO) für eine Anweisung ermitteln

Einige Anweisungen (z. B. mathematische Anweisungen und Übertragungsanweisungen ) zeigen Parameter für EN und ENO an. Diese Parameter beziehen sich auf den Signalfluss in KOP oder FUP und legen fest, ob die Anweisung in diesem Zyklus ausgeführt wird. In SCL können Sie den Parameter ENO auch für einen Codebaustein angeben.

- EN (Freigabeeingang) ist ein Boolescher Eingang. An diesem Eingang muss Signalfluss (EN = 1) vorhanden sein, damit die Box ausgeführt werden kann. Wenn der Eingang EN einer KOP-Box direkt an die linke Stromschiene angeschlossen ist, wird die Anweisung immer ausgeführt.
- ENO (Freigabeausgang) ist ein Boolescher Ausgang. Liegt am Eingang EN einer Box ein Signalfluss an und die Box wird fehlerfrei ausgeführt, dann leitet der Ausgang ENO den Signalfluss (ENO = 1) zum nächsten Element weiter. Tritt während der Ausführung der Box ein Fehler auf, dann wird der Signalfluss an der Box-Anweisung, die den Fehler verursacht hat, beendet (ENO = 0).

Tabelle 6- 3 Operanden für EN und ENO

Programm-Editor	Eingänge/Ausgänge	Operanden	Datentyp
KOP	EN, ENO	Signalfluss	Bool
FUP	EN	E, E:P, A, M, DB, Temp, Signalfluss	Bool
	ENO	Signalfluss	Bool
SCL	EN <sup>1</sup>	WAHR, FALSCH	Bool
	ENO <sup>2</sup>	WAHR, FALSCH	Bool

<sup>1</sup> Die Verwendung von EN steht nur bei FBs zur Verfügung.

<sup>2</sup> Die Verwendung von ENO mit dem SCL-Codebaustein ist optional. Sie müssen die SCL-Übersetzung so einrichten, dass ENO bei Fertigstellung der Codebausteinbearbeitung gesetzt wird.

## Setzen von ENO in SCL konfigurieren

Um die SCL-Übersetzung so einzurichten, dass ENO gesetzt wird, gehen Sie folgendermaßen vor:

1. Wählen Sie im Menü "Extras" den Befehl "Einstellungen".
2. Erweitern Sie die Eigenschaften "PLC-Programmierung" und wählen Sie "SCL (Structured Control Language)".
3. Wählen Sie die Option "ENO automatisch setzen".

## Auswirkung der Parameter Ret\_Val oder Status auf ENO

Einige Anweisungen, z. B. die Kommunikationsanweisungen oder die Anweisungen für die Zeichenkettenkonvertierung, verfügen über einen Ausgangsparameter, der Informationen über die Verarbeitung der Anweisung enthält. Einige Anweisungen haben beispielsweise den Parameter Ret\_Val (Rückgabewert), üblicherweise vom Datentyp Int, der Statusinformationen im Bereich von -32768 bis +32767 enthält. Andere Anweisungen haben den Parameter Status, typischerweise vom Datentyp Word, der Statusinformationen im Hexadezimalbereich von 16#0000 bis 16#FFFF speichert. Der numerische Wert in einem der Parameter Ret\_Val oder Status gibt den Zustand von ENO der jeweiligen Anweisung an.

- Ret\_Val: Ein Wert zwischen 0 und 32767 setzt ENO typischerweise = 1 (bzw. WAHR). Ein Wert zwischen -32768 und -1 setzt ENO typischerweise = 0 (bzw. FALSCH). Um Ret\_Val auszuwerten, ändern Sie die Darstellung in hexadezimal.
- Status: Ein Wert zwischen 16#0000 und 16#7FFF setzt ENO typischerweise = 1 (bzw. WAHR). Ein Wert zwischen 16#8000 und 16#FFFF setzt ENO typischerweise = 0 (bzw. FALSCH).

Anweisungen, die über mehrere Zyklen ausgeführt werden, haben häufig den Parameter Busy (Bool), um kenntlich zu machen, dass die Anweisung aktiv ist, die Ausführung jedoch noch nicht beendet ist. Solche Anweisungen haben auch häufig auch den Parameter Done (Bool) und den Parameter Error (Bool). Done zeigt an, dass die Anweisung fehlerfrei ausgeführt wurde, und Error weist darauf hin, dass die Anweisung mit Fehler beendet wurde.

- Wenn Busy = 1 (bzw. WAHR), ist ENO = 1 (bzw. WAHR).
- Wenn Done = 1 (bzw. WAHR), ist ENO = 1 (bzw. WAHR).
- Wenn Error = 1 (bzw. WAHR), ist ENO = 0 (bzw. FALSCH).

## Siehe auch

Operationen OK und NOT\_OK (Seite 208)

## 6.6 Schutz

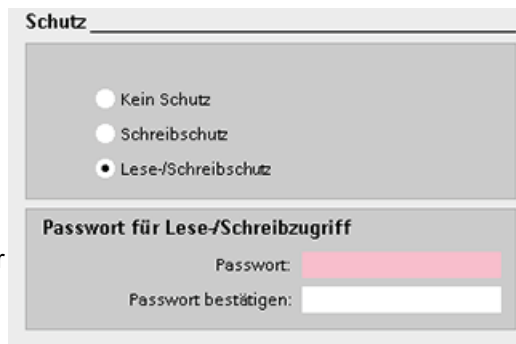
### 6.6.1 Zugriffsschutz für die CPU

Die CPU bietet drei Sicherheitsstufen, um den Zugang zu bestimmten Funktionen einzuschränken. Mit dem Einrichten der Schutzstufe und des Passworts für eine CPU schränken Sie die Funktionen und Speicherbereiche ein, die ohne Eingabe eines Passworts zugänglich sind.

Die Groß- und Kleinschreibung des Passwortes ist zu beachten.

Um das Passwort zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der "Gerätekonfiguration" die CPU aus.
2. Wählen Sie im Inspektorfenster das Register "Eigenschaften".
3. Wählen Sie die Eigenschaft "Schutz", um die Schutzstufe auszuwählen und ein Passwort einzugeben.



Jede Schutzstufe lässt auch ohne Eingabe eines Passworts den uneingeschränkten Zugriff auf bestimmte Funktionen zu. Die Voreinstellung der CPU ist "ohne Einschränkung" und "ohne Passwortschutz". Um den Zugang zu einer CPU zu schützen, müssen Sie die Eigenschaften der CPU einrichten und das Passwort eingeben.

Wenn Sie ein Netzpasswort eingeben, dann wirkt sich dieses Passwort nicht auf den Passwortschutz der CPU aus. Der Passwortschutz gilt nicht für die Ausführung der Anweisungen des Anwenderprogramms einschließlich Kommunikationsfunktionen. Die Eingabe des richtigen Passworts gestattet den ungehinderten Zugriff auf alle Funktionen.

Die Kommunikation zwischen CPUs (über die Kommunikationsfunktionen in den Codebausteinen) wird durch die Schutzstufe der CPU nicht eingeschränkt. Auch die HMI-Funktionalität bleibt uneingeschränkt erhalten.

Tabelle 6- 4 Schutzstufen der CPU

Schutzstufe	Zugangsbeschränkungen
Kein Schutz	Ungehinderter Zugang ohne Passwortschutz.
Schreibschutz	HMI-Zugang und ungehinderte Kommunikation zwischen CPUs ohne Passwortschutz. Ein Passwort ist für Änderungen (Schreibzugriffe) in der CPU und für den Wechsel des Betriebszustands der CPU (RUN/STOP) erforderlich.
Lese-/Schreibschutz	HMI-Zugang und ungehinderte Kommunikation zwischen CPUs ohne Passwortschutz. Ein Passwort ist zum Lesen der Daten in der CPU, für Änderungen (Schreiben) in der CPU und für den Wechsel des Betriebszustands der CPU (RUN/STOP) erforderlich.



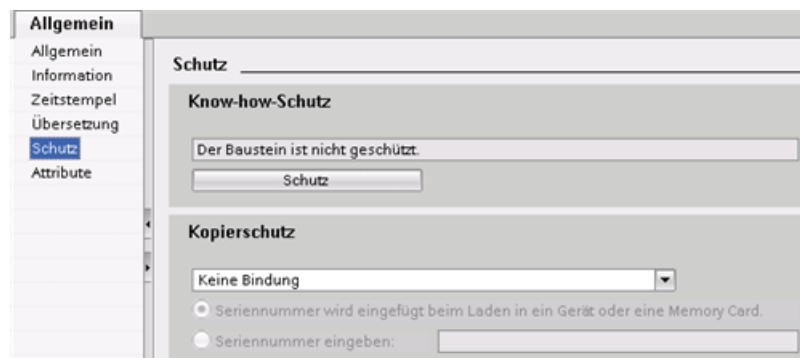
## 6.6.2 Knowhow-Schutz

Mit dem Knowhow-Schutz können Sie einen oder mehrere Codebausteine (OB, FB, FC oder DB) in Ihrem Programm vor unbefugtem Zugriff schützen. Sie können ein Passwort eingeben, um den Zugriff auf einen Codebaustein einzuschränken. Der Passwortschutz verhindert das unbefugte Lesen oder Ändern des Codebausteins. Ohne Passwort können nur die folgenden Informationen zum Codebaustein gelesen werden:

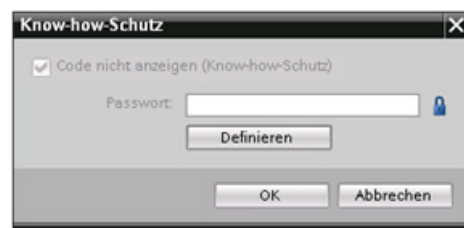
- Bausteintitel, Kommentar und Bausteineigenschaften
- Übertragungsparameter (IN, OUT, IN\_OUT, Rückgabe)
- Aufrufstruktur des Programms
- Globale Variablen in den Querverweisen (ohne Information über die Verwendung), lokale Variablen sind jedoch verborgen

Wenn Sie einen Baustein für den Knowhow-Schutz konfigurieren, so ist der Code in diesem Baustein erst nach Eingabe des Passworts zugänglich.

Den Knowhow-Schutz eines Codebausteins konfigurieren Sie in der Taskcard "Eigenschaften" des jeweiligen Codebausteins. Nach dem Öffnen des Codebausteins wählen Sie unter "Eigenschaften" die Option "Schutz".



1. In den Eigenschaften des Codebausteins klicken Sie auf die Schaltfläche "Eigenschaften", um den Dialog "Knowhow-Schutz" anzuzeigen.
2. Klicken Sie auf die Schaltfläche "Definieren", um das Passwort einzugeben.



Nachdem Sie das Passwort eingegeben und bestätigt haben, klicken Sie auf "OK".

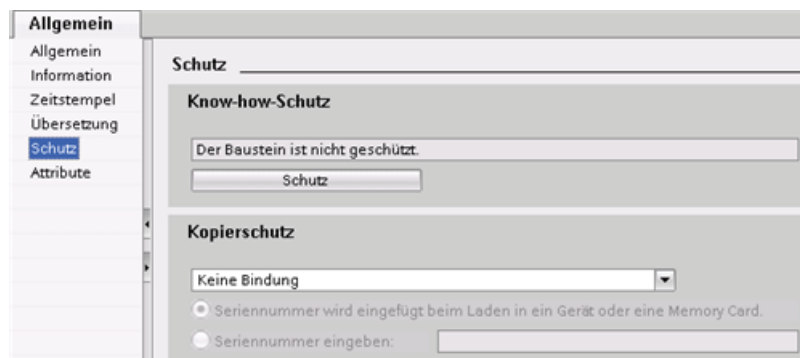


### 6.6.3 Kopierschutz

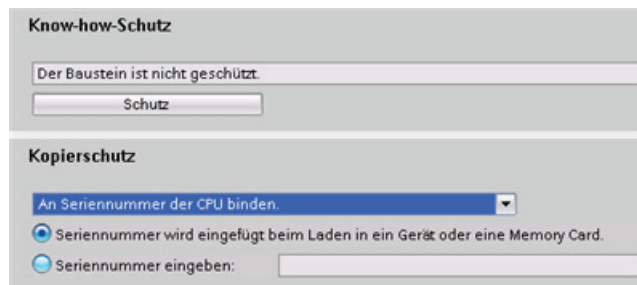
Eine weitere Sicherheitsfunktion ermöglicht Ihnen, das Programm oder die Codebausteine mit einer bestimmten Memory Card oder CPU zu verknüpfen. Diese Funktion ist vor allem zum Schutz geistigen Eigentums nützlich. Wird ein Programm oder ein Baustein mit einem bestimmten Gerät verknüpft, so ist die Verwendung dieses Programms oder dieses Bausteins nur in Verbindung mit einer bestimmten Memory Card oder CPU möglich. Mit dieser Funktion kann ein Programm oder ein Codebaustein elektronisch (zum Beispiel über Internet oder E-Mail) oder durch Versenden eines Speichermoduls verschickt werden.

Um einen Baustein mit einer bestimmten CPU oder Memory Card zu verknüpfen, öffnen Sie die Taskcard "Eigenschaften" des jeweiligen Codebausteins.

1. Nach dem Öffnen des Codebausteins wählen Sie "Schutz".



2. Wählen Sie in der Klappliste "Kopierschutz" die Option aus, um den Codebaustein mit einer Memory Card oder einer bestimmten CPU zu verknüpfen.



3. Wählen Sie die Art des Kopierschutzes aus und geben Sie die Seriennummer der Memory Card oder CPU ein.

---

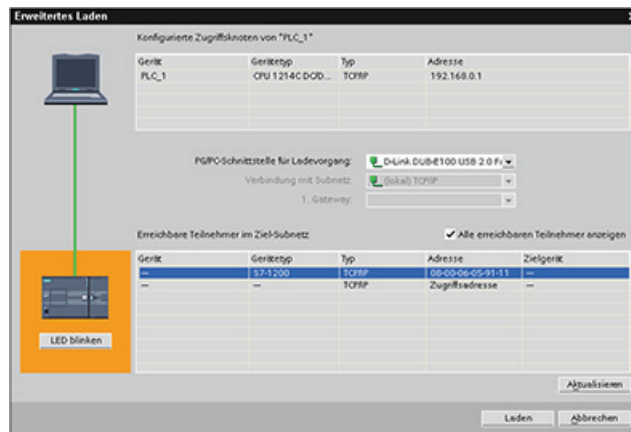
#### Hinweis

Bei der Seriennummer ist die Groß- und Kleinschreibung zu beachten.

---

## 6.7 Laden der Programmelemente

Sie können die Elemente Ihres Projekts aus dem Programmiergerät in die CPU laden. Wenn Sie ein Projekt laden, speichert die CPU das Anwenderprogramm (OBs, FCs, FBs und DBs) im nullspannungsfesten Speicher.



Sie können Ihr Projekt aus dem Programmiergerät in die CPU laden, und zwar von einer der folgenden Stellen:

- Projektnavigation: Klicken Sie mit der rechten Maustaste auf das Programmelement und wählen Sie dann "Laden in CPU".
- Onlinemenü: Klicken Sie auf die Option "Laden in Gerät".
- Symbolleiste: Klicken Sie auf das Symbol "Laden in Gerät".

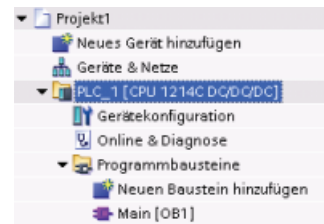
## 6.8 Laden aus der CPU

### 6.8.1 Elemente des Projekts kopieren

Sie können die Programmbausteine einer Online-CPU oder einer an Ihr Programmiergerät angeschlossenen Memory Card auch kopieren.

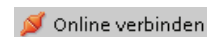
Bereiten Sie das Offline-Projekt für die kopierten Programmbausteine vor:

1. Fügen Sie eine CPU hinzu, die der Online-CPU entspricht.
2. Erweitern Sie den CPU-Knoten, so dass der Ordner "Programmbausteine" angezeigt wird.



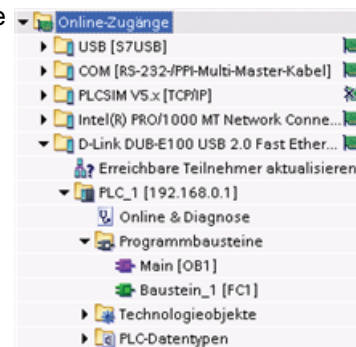
Um die Programmbausteine aus der Online-CPU in das Offline-Projekt zu laden, gehen Sie folgendermaßen vor:

1. Klicken Sie im Offline-Projekt auf den Ordner "Programmbausteine".
2. Klicken Sie auf die Schaltfläche "Online verbinden".
3. Wählen Sie die Schaltfläche "Laden aus CPU".
4. Bestätigen Sie den Vorgang im Dialog Laden aus CPU (Seite 719).

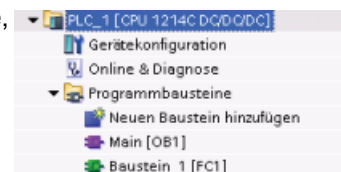


Als Alternative zur vorherigen Vorgehensweise gehen Sie folgendermaßen vor:

1. Erweitern Sie in der Projektnavigation den Knoten "Online-Zugänge", um die Programmbausteine in der Online-CPU auszuwählen.
2. Erweitern Sie den Knoten für das Netzwerk und doppelklicken Sie auf "Erreichbare Teilnehmer aktualisieren".
3. Erweitern Sie den Knoten für die CPU.
4. Ziehen Sie den Ordner "Programmbausteine" mit der Maus von der Online-CPU in den Ordner "Programmbausteine" Ihres Offline-Projekts.
5. Aktivieren Sie im Dialog "Vorschau für das Laden von Gerät" das Kontrollkästchen "Fortfahren", und klicken Sie dann auf "Laden von Gerät".



Nach dem Ladevorgang werden alle Programmbausteine, Technologiebausteine und Variablen im Offline-Bereich angezeigt.



### Hinweis

Sie können die Programmbausteine aus der Online-CPU in ein vorhandenes Programm kopieren. Der Ordner "Programmbausteine" des Offline-Projekts muss nicht leer sein. Das vorhandene Programm wird jedoch gelöscht und durch das Anwenderprogramm aus der Online-CPU ersetzt.

### 6.8.2 Mit der Vergleichsfunktion arbeiten

Mit dem Editor "Vergleichen" (Seite 725) in STEP 7 ermitteln Sie Unterschiede zwischen dem Online- und dem Offline-Projekt. Dies kann vor dem Laden des Programms aus der CPU nützlich sein.

## 6.9 Debugging und Testen des Programms

### 6.9.1 Daten in der CPU beobachten und steuern

Sie können Werte in der Online-CPU beobachten und steuern (siehe folgende Tabelle).

Tabelle 6- 5 Daten mit STEP 7 beobachten und steuern

Editor	Beobachten	Steuern	Forcen
Beobachtungstabelle	Ja	Ja	Nein
Forcetabelle	Ja	Nein	Ja
Programmiereditor	Ja	Ja	Nein
Variablentabelle	Ja	Nein	Nein
DB-Editor	Ja	Nein	Nein



In einer Beobachtungstabelle beobachten



Im KOP-Editor beobachten

Im Kapitel "Online und Diagnose" finden Sie weitere Informationen zum Thema Daten in der CPU beobachten und steuern (Seite 726).

## 6.9.2 Beobachtungstabellen und Forcetabellen

Mit Hilfe von "Beobachtungstabellen" können Sie die Werte eines Anwenderprogramms, das von der Online-CPU ausgeführt wird, überwachen und ändern. Sie können in Ihrem Projekt unterschiedliche Beobachtungstabellen erstellen und speichern, um eine Vielzahl von Testumgebungen abzudecken. So können Sie Tests zum Beispiel bei der Inbetriebnahme oder für Service- und Wartungszwecke durchführen.

Mit einer Beobachtungstabelle können Sie die Ausführung des Anwenderprogramms durch die CPU überwachen und in die Ausführung eingreifen. Sie können nicht nur für die Variablen der Codebausteine und Datenbausteine, sondern auch für die Speicherbereiche der CPU, einschließlich Eingänge und Ausgänge (E und A), periphere Eingänge (E:P), Merker (M) und Datenbausteine (DB) Werte aufrufen und ändern.

Mit der Beobachtungstabelle können Sie die physischen Ausgänge (A:P) einer CPU, die sich im Betriebszustand STOP befindet, freigeben. Beispielsweise können Sie den Ausgängen bestimmte Werte zuweisen, während Sie die Verdrahtung der CPU testen.

STEP 7 bietet zudem eine Forcetabelle zum Forcen einer Variablen auf einen bestimmten Wert. Weitere Informationen zum Forcen finden Sie im Abschnitt zum Forcen von Werten in der CPU (Seite 733) im Kapitel "Online und Diagnose".

---

### Hinweis

Die Forcewerte werden in der CPU und nicht in der Beobachtungstabelle gespeichert.

Sie können keinen Eingang forcen (Adresse "E"). Sie können jedoch einen Peripherieeingang forcen. Um einen Peripherieeingang zu forcen, hängen Sie ein ":P" an die Adresse an (Beispiel: "On:P").

---

## 6.9.3 Querverweis zum Anzeigen der Verwendung

Das Inspektorfenster zeigt Querverweise dazu an, wie ein Objekt innerhalb des gesamten Projekts verwendet wird, z. B. im Anwenderprogramm, in der CPU oder den HMI-Geräten. Im Register "Querverweis" werden die Instanzen angezeigt, wo und von welchen anderen Objekten ein ausgewähltes Objekt verwendet wird. Das Inspektorfenster enthält außerdem Bausteine, die nur online innerhalb der Querverweise verfügbar sind. Um die Querverweise anzuzeigen, wählen Sie den Befehl "Querverweise anzeigen". (In der Projektansicht befindet sich dieser Befehl im Menü "Werkzeuge".)

---

### Hinweis

Zum Anzeigen der Querverweisinformationen muss der Editor nicht geschlossen werden.

---

Die Einträge der Querverweisliste können verschieden sortiert werden. Die Liste der Querverweise bietet einen Überblick über die Verwendung von Speicheradressen und Variablen im Anwenderprogramm.

- Wenn Sie ein Programm anlegen oder ändern, behalten Sie einen Überblick über die verwendeten Operanden, Variablen und Bausteinaufrufe.
- Aus den Querverweisen können Sie direkt an die Stelle springen, an der die Operanden und Variablen verwendet werden.
- Während eines Programmtests oder einer Fehlerbehebung erhalten Sie Informationen dazu, welche Speicheradresse von welchem Befehl in welchem Baustein verarbeitet wird, welche Variable in welchem Bild verwendet wird und welcher Baustein von welchem anderen Baustein aufgerufen wird.

Tabelle 6- 6 Querverweiselemente

Spalte	Beschreibung
Objekt	Name des Objekts, das die angegebenen unterlagerten Objekte verwendet oder das von den unterlagerten Objekten verwendet wird
Anzahl	Anzahl Verwendungen
Adresse	Der Ort der Verwendung, z. B. ein Netzwerk
Eigenschaft	Besondere Eigenschaften der referenzierten Objekte, z. B. die Variablennamen in Multiinstanz-Deklarationen
als	Zeigt zusätzliche Informationen zum Objekt an, z. B. ob ein Instanz-DB als Vorlage oder als Multiinstanz verwendet wird
Zugriff	Art des Zugriffs, d. h. ob auf den Operanden Lesezugriff (R) und/oder Schreibzugriff (W) besteht
Adresse	Adresse des Operanden
Typ	Angabe, mit welchem Typ und welcher Sprache das Objekt angelegt wurde
Pfad	Pfad des Objekts in der Projektnavigation

## 6.9.4 Aufrufstruktur zur Prüfung der Aufrufhierarchie

Die Aufrufstruktur zeigt die Aufrufhierarchie des Bausteins innerhalb Ihres Anwenderprogramms. Sie bietet einen Überblick über die verwendeten Bausteine, die Aufrufe anderer Bausteine, die Beziehungen zwischen Bausteinen, die Datenanforderungen an jeden Baustein sowie den Status der einzelnen Bausteine. Die Bausteine in der Aufrufstruktur können mit dem Programmiereditor geöffnet und bearbeitet werden.

Durch Anzeigen der Aufrufstruktur erhalten Sie eine Liste der im Anwenderprogramm verwendeten Bausteine. STEP 7 zeigt die erste Ebene der Aufrufstruktur hervorgehoben an und zeigt auch die Bausteine an, die durch keinen anderen Baustein im Programm aufgerufen werden. Die erste Ebene der Aufrufstruktur enthält die OBs sowie diejenigen FCs, FBs und DBs, die nicht durch einen OB aufgerufen werden. Von anderen Bausteinen aufgerufene Codebausteine erscheinen eingerückt unter dem aufrufenden Baustein. In der Aufrufstruktur werden nur die Bausteine angezeigt, die von einem Codebaustein aufgerufen werden.

Sie können selektiv nur jene Bausteine anzeigen, die innerhalb der Aufrufstruktur Konflikte verursachen. Folgende Bedingungen führen zu Konflikten:

- Bausteine, die Aufrufe mit älteren oder neueren Zeitstempeln im Code ausführen
- Bausteine, die einen Baustein mit geänderter Schnittstelle aufrufen
- Bausteine, die eine Variable mit geänderter Adresse und/oder geändertem Datentyp verwenden
- Bausteine, die weder direkt noch indirekt durch einen OB aufgerufen werden
- Bausteine, die einen nicht vorhandenen oder fehlenden Baustein aufrufen

Sie können mehrere Bausteinaufrufe und Datenbausteine zu einer Gruppe zusammenfassen. Über eine Klappliste können Sie die Verknüpfungen mit den verschiedenen Aufrufstellen anzeigen.

Sie können außerdem eine Konsistenzprüfung durchführen, um Zeitstempelkonflikte aufzuzeigen. Zeitstempelkonflikte können durch die Änderung des Zeitstempels eines Bausteins während oder nach der Programmgenerierung verursacht werden. Diese Konflikte führen zu Inkonsistenzen zwischen den aufrufenden und den aufgerufenen Bausteinen.

- Die meisten Zeitstempel- und Schnittstellenkonflikte lassen sich durch erneutes Übersetzen der Codebausteine beheben.
- Wenn durch Übersetzen die Inkonsistenzen nicht beseitigt werden, navigieren Sie mit dem Programmierer über die Verknüpfung in der Spalte "Details" zur Quelle des Problems. Hier können Sie die Inkonsistenzen manuell beseitigen.
- Sind Bausteine rot markiert, müssen sie erneut übersetzt werden.



## Anweisungen

### 7.1 Bitverknüpfung

#### 7.1.1 Bitverknüpfungskontakte und -spulen

KOP und FUP verarbeiten Boolesche Logik sehr effektiv. SCL ist zwar besonders effektiv bei komplexen mathematischen Berechnungen und bei Projektsteuerstrukturen, doch Sie können SCL auch für Boolesche Logik verwenden.

#### KOP-Kontakte

Tabelle 7- 1 Schließer- und Öffnerkontakte



KOP	SCL	Beschreibung
"IN" 	<pre>IF in THEN     Statement; ELSE     Statement; END IF;</pre>	Schließer- und Öffnerkontakte: Sie können Kontakte untereinander verschalten und so Ihre eigene Verschaltungslogik erstellen. Nutzt das von Ihnen angegebene Eingangsbit die Speicherkennung E (Eingang) oder A (Ausgang), so wird der Bitwert aus dem Prozessabbildregister gelesen. Die physischen Kontaktsignale in Ihrem Steuerungsprozess werden mit Eingangsanschlüssen der PLC-Geräts verschaltet. Die CPU fragt die verschalteten Eingangssignale ab und aktualisiert fortlaufend die entsprechenden Zustandswerte im Prozessabbild der Eingänge.
"IN" 	<pre>IF NOT (in) THEN     Statement; ELSE     Statement; END_IF;</pre>	Das direkte Lesen eines physischen Eingangs geben Sie mit ":P" nach der E-Adresse an (Beispiel: "%I3.4:P"). Beim direkten Lesen werden die Bitdatenwerte direkt aus dem physischen Eingang und nicht aus dem Prozessabbild gelesen. Beim direkte Lesen wird das Prozessabbild nicht aktualisiert.

Tabelle 7- 2 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Bool	Zugewiesenes Bit


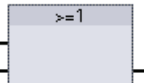
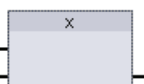
- Der Schließer ist geschlossen (EIN), wenn der zugewiesene Bitwert gleich 1 ist.
- Der Öffner ist geschlossen (EIN), wenn der zugewiesene Bitwert gleich 0 ist.
- In Reihe geschaltete Kontakte bilden logische UND-Verknüpfungen.
- Parallel geschaltete Kontakte bilden logische ODER-Verknüpfungen.

**UND-, ODER- und XOR-Boxen in FUP**

Bei der FUP-Programmierung werden Netzwerke mit KOP-Kontakten in die Box-Netzwerke UND (&), ODER (>=1) und Exklusiv ODER (x) umgewandelt, in denen Sie Bitwerte für die Ein- und -Ausgänge der Box angeben können. Sie können ferner Verschaltungen mit anderen Logik-Boxen herstellen und so Ihre eigene Verschaltungslogik erstellen. Nachdem die Box in Ihrem Netzwerk platziert ist, können Sie die Funktion "Eingang einfügen" aus der Funktionsleiste "Favoriten" oder dem Anweisungsverzeichnis zur Eingangsseite der Box ziehen, um weitere Eingänge hinzuzufügen. Sie können auch mit der rechten Maustaste auf den Eingangsanschluss der Box klicken und "Eingang einfügen" auswählen.

Die Ein- und Ausgänge einer Box können mit anderen Boxen verschaltet werden oder Sie können eine Bitadresse oder einen Bitsymbolnamen für einen unverschalteten Eingang eingeben. Bei der Ausführung der Box-Anweisung werden die Eingangszustände auf die binäre Box-Verknüpfung geschaltet und dann der Box-Ausgang, sofern zutreffend, auf Wahr gesetzt.

Tabelle 7- 3 Boxen UND, ODER und EXKLUSIV ODER

FUP	SCL <sup>1</sup>	Beschreibung
 <p>"IN1" "IN2"</p>	<pre>out := in1 AND in2;</pre>	Damit der Ausgang WAHR ist, müssen alle Eingänge einer UND-Box WAHR sein
 <p>"IN1" "IN2"</p>	<pre>out := in1 OR in2;</pre>	Damit der Ausgang WAHR ist, muss ein beliebiger Eingang einer ODER-Box WAHR sein
 <p>"IN1" "IN2"</p>	<pre>out := in1 XOR in2;</pre>	Damit der Ausgang WAHR ist, muss eine ungerade Anzahl der Eingänge einer XOR-Box WAHR sein

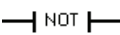
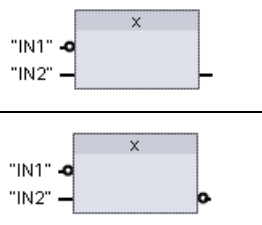
<sup>1</sup> In SCL: Sie müssen das Ergebnis der Operation einer Variable zuweisen, damit es in einer anderen Anweisung verwendet werden kann.

Tabelle 7- 4 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN1, IN2	Bool	Eingangsbit

## Logikinvertierer NOT

Tabelle 7-5 Logikinvertierer NOT

KOP	FUP	SCL	Beschreibung
		NOT	<p>Bei der FUP-Programmierung können Sie die Funktion "Binäreingang invertieren" aus der Funktionsleiste "Favoriten" oder dem Anweisungsverzeichnis auf einen Eingang oder einen Ausgang ziehen, um einen Logikinvertierer für diesen Box-Anschluss zu erstellen.</p> <p>Der KOP-Kontakt NOT invertiert den logischen Zustand des Signalfusseingangs.</p> <ul style="list-style-type: none"> <li>Ist kein Signalfluss zum NOT-Kontakt vorhanden, so steht ein Signalfluss am Ausgang an.</li> <li>Ist ein Signalfluss zum NOT-Kontakt vorhanden, so steht kein Signalfluss am Ausgang an.</li> </ul>

## Ausgangsspule und Zuweisungsbox

Die Anweisung für den Spulenausgang schreibt einen Wert in ein Ausgangsbit. Nutzt das angegebene Ausgangsbit die Speicherkennung A, so schaltet die CPU das Ausgangsbit im Prozessabbildregister ein oder aus und setzt das angegebene Bit jeweils entsprechend dem Signalfluss. Die Ausgangssignale für Ihre Steuerstellglieder werden mit den Ausgangsklemmen der CPU verschaltet. Im Betriebszustand RUN fragt die CPU fortlaufend die Eingangssignale ab, verarbeitet die Eingangszustände gemäß der Programmlogik und reagiert dann, indem sie die neuen Ausgangswerte im Prozessabbild der Ausgänge setzt. Nach jeder Programmausführung überträgt die CPU die im Prozessabbildregister gespeicherte neue Reaktion auf den Ausgangszustand zu den verschalteten Ausgangsklemmen.

Tabelle 7-6 Ausgangsspule (KOP) und Ausgangszuweisungsbox (FUP)

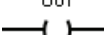

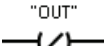
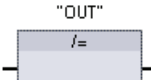
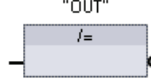
KOP	FUP	SCL	Beschreibung
		<pre>out := &lt;Boolescher Ausdruck&gt;;</pre>	<p>Bei der FUP-Programmierung werden die KOP-Spulen in Zuweisungsboxen (= und /=) umgewandelt, wobei eine Bitadresse für den Boxausgang anzugeben ist. Die Ein- und Ausgänge einer Box können mit anderen Boxverknüpfungen verschaltet werden oder Sie können eine Bitadresse eingeben.</p> <p>Das direkte Schreiben eines physischen Ausgangs geben Sie mit ":P" nach der A-Adresse an (Beispiel: "%Q3.4:P"). Beim direkten Schreiben werden die Bitdatenwerte in den Ausgang im Prozessabbild und direkt in den physischen Ausgang geschrieben.</p>
		<pre>out := NOT &lt;Boolescher Ausdruck&gt;;</pre>	
			

Tabelle 7-7 Datentypen für die Parameter

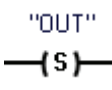
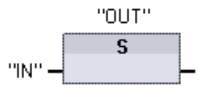
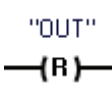
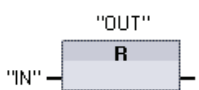
Parameter	Datentyp	Beschreibung
OUT	Bool	Zugewiesenes Bit

- Ist ein Signalfluss durch eine Ausgangsspule vorhanden oder eine FUP-Box "=" aktiviert, so wird das Ausgangsbit auf 1 gesetzt.
- Ist kein Signalfluss durch eine Ausgangsspule vorhanden oder keine FUP-Box "=" aktiviert, so wird das Ausgangsbit auf 0 gesetzt.
- Ist ein Signalfluss durch eine invertierte Ausgangsspule vorhanden oder eine FUP-Box "/=" aktiviert, so wird das Ausgangsbit auf 0 gesetzt.
- Ist kein Signalfluss durch eine invertierte Ausgangsspule vorhanden oder keine FUP-Box "/=" aktiviert, so wird das Ausgangsbit auf 1 gesetzt.

### 7.1.2 Setz- und Rücksetzoperationen

#### 1 Bit setzen und rücksetzen

Tabelle 7-8 Anweisungen S und R

KOP	FUP	SCL	Beschreibung
		Nicht verfügbar	Wird S (Setzen) aktiviert, wird der Datenwert an Adresse OUT auf 1 gesetzt. Wird S nicht aktiviert, verändert sich OUT nicht.
		Nicht verfügbar	Wird R (Rücksetzen) aktiviert, wird der Datenwert an Adresse OUT auf 0 gesetzt. Wird R nicht aktiviert, verändert sich OUT nicht.

- <sup>1</sup> In KOP und FUP: Diese Anweisungen können an jeder beliebigen Stelle im Netzwerk eingefügt werden.
- <sup>2</sup> In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 7-9 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN (oder Verschalten mit Kontakt-/Steuerungslogik)	Bool	Zu beobachtende Bitadresse
OUT	Bool	Zu setzende oder zurückzusetzende Bitadresse

## Bitfeld setzen und rücksetzen

Tabelle 7- 10 Anweisungen SET\_BF und RESET\_BF

KOP <sup>1</sup>	FUP	SCL	Beschreibung
<pre>"OUT" └─(SET_BF)─┘   "n"</pre>	<pre>"OUT" ┌─ SET_BF ─┐ │ EN        │ └─ N        │</pre>	Nicht verfügbar	Wird SET_BF aktiviert, wird der Datenwert 1 in "n" Bits geschrieben, beginnend an Adresse OUT. Wird SET_BF nicht aktiviert, verändert sich OUT nicht.
<pre>"OUT" └─(RESET_BF)─┘   "n"</pre>	<pre>"OUT" ┌─ RESET_BF ─┐ │ EN         │ └─ N         │</pre>	Nicht verfügbar	RESET_BF schreibt den Datenwert 0 in "n" Bits, beginnend an Adresse OUT. Wird RESET_BF nicht aktiviert, verändert sich OUT nicht.

- 1 In KOP und FUP: Diese Anweisungen müssen ganz rechts in einer Verzweigung angeordnet werden.
- 2 In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 7- 11 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
OUT	Bool	Startelement des Bitfelds, das gesetzt oder zurückgesetzt werden soll (Beispiel: #MyArray[3])
n	Konstante (UInt)	Anzahl der zu schreibenden Bits

## Flipflop-Schaltung vorrangig setzen und vorrangig rücksetzen

Tabelle 7- 12 Anweisungen RS und SR

KOP/FUP	SCL	Beschreibung
<pre>"OUT" ┌─ RS ─┐ │ R     │ │ S1    │ └─ Q ─┘</pre>	Nicht verfügbar	RS ist ein Flipflop, bei dem das Setzen Vorrang hat. Sind beide Signale Setzen (S1) und Rücksetzen (R) wahr, ist die Ausgangsadresse OUT 1.
<pre>"OUT" ┌─ SR ─┐ │ S     │ │ R1    │ └─ Q ─┘</pre>	Nicht verfügbar	SR ist ein Flipflop, bei dem das Rücksetzen Vorrang hat. Sind beide Signale Setzen (S) und Rücksetzen (R1) wahr, ist die Ausgangsadresse OUT 0.

- 1 In KOP und FUP: Diese Anweisungen müssen ganz rechts in einer Verzweigung angeordnet werden.
- 2 In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 7- 13 Datentypen für die Parameter

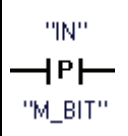
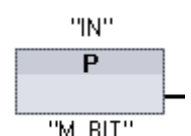
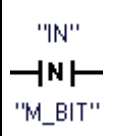
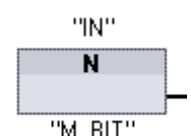
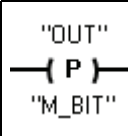
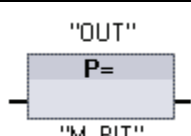
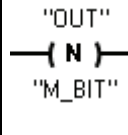
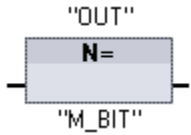
Parameter	Datentyp	Beschreibung
S, S1	Bool	Eingang setzen; 1 weist auf Dominanz hin
R, R1	Bool	Eingang zurücksetzen; 1 weist auf Dominanz hin
OUT	Bool	Zugewiesener Bitausgang "OUT"
Q	Bool	Folgt dem Zustand von Bit "OUT"

Der Parameter OUT gibt die Bitadresse an, die gesetzt bzw. zurückgesetzt wird. Der optionale OUT-Ausgang Q gibt den Signalzustand der Adresse OUT an.

Anweisung	S1	R	Bit "OUT"
RS	0	0	Vorheriger Zustand
	0	1	0
	1	0	1
	1	1	1
SR	<b>S</b>	<b>R1</b>	
	0	0	Vorheriger Zustand
	0	1	0
	1	0	1
	1	1	0



### 7.1.3 Operationen Steigende Flanke und Fallende Flanke

Tabelle 7- 14 Anweisungen "Steigende Flanke" und "Fallende Flanke"

KOP	FUP	SCL	Beschreibung
		Nicht verfügbar	<p>KOP: Der Zustand dieses Kontakts ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am zugewiesenen Bit IN erkannt wird. Der logische Zustand des Kontakts wird dann mit dem Eingangszustand des Signalfusses verknüpft, um den Ausgangszustand des Signalfusses zu setzen. Der Kontakt P kann an einer beliebigen Stelle im Netzwerk angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p> <p>FUP: Der logische Zustand des Ausgangs ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am zugewiesenen Eingangsbit erkannt wird. Die Box P darf nur am Anfang einer Verzweigung angeordnet werden.</p>
		Nicht verfügbar	<p>KOP: Der Zustand dieses Kontakts ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am zugewiesenen Eingangsbit erkannt wird. Der logische Zustand des Kontakts wird dann mit dem Eingangszustand des Signalfusses verknüpft, um den Ausgangszustand des Signalfusses zu setzen. Der Kontakt N kann an einer beliebigen Stelle im Netzwerk angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p> <p>FUP: Der logische Zustand des Ausgangs ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am zugewiesenen Eingangsbit erkannt wird. Die Box N darf nur am Anfang einer Verzweigung angeordnet werden.</p>
		Nicht verfügbar	<p>KOP: Das zugewiesene Bit OUT ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am Signalfuss, der in die Spule eintritt, erkannt wird. Der Eingangszustand des Signalfusses durchläuft die Spule immer als Ausgangszustand des Signalfusses. Die Spule P kann an jeder beliebigen Stelle im Netzwerk eingefügt werden.</p> <p>FUP: Das zugewiesene Bit OUT ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am logischen Zustand des Eingangsanschlusses der Box oder am zugewiesenen Eingangsbit erkannt wird, sofern sich die Box am Anfang einer Verzweigung befindet. Der logische Zustand des Eingangs durchläuft die Box immer als logischer Zustand des Ausgangs. Die Box P= kann an jeder beliebigen Stelle in der Verzweigung eingefügt werden.</p>
		Nicht verfügbar	<p>KOP: Das zugewiesene Bit OUT ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am Signalfuss, der in die Spule eintritt, erkannt wird. Der Eingangszustand des Signalfusses durchläuft die Spule immer als Ausgangszustand des Signalfusses. Die Spule N kann an jeder beliebigen Stelle im Netzwerk eingefügt werden.</p> <p>FUP: Das zugewiesene Bit OUT ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am logischen Zustand des Eingangsanschlusses der Box oder am zugewiesenen Eingangsbit erkannt wird, sofern sich die Box am Anfang einer Verzweigung befindet. Der logische Zustand des Eingangs durchläuft die Box immer als logischer Zustand des Ausgangs. Die Box N= kann an jeder beliebigen Stelle in der Verzweigung eingefügt werden.</p>

<sup>1</sup> In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 7- 15 Anweisungen P\_TRIG und N\_TRIG

KOP/FUP	SCL	Beschreibung
	Nicht verfügbar	<p>Der Signalfluss oder der logische Zustand des Ausgangs Q ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am Eingangszustand CLK (FUP) oder am Signalfluss CLK (KOP) erkannt wird.</p> <p>In KOP darf die Anweisung P_TRIG nicht am Anfang oder Ende eines Netzwerks angeordnet werden. In FUP darf die Anweisung P_TRIG an einer beliebigen Stelle angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p>
	Nicht verfügbar	<p>Der Signalfluss oder der logische Zustand des Ausgangs Q ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am Eingangszustand CLK (FUP) oder am Signalfluss CLK (KOP) erkannt wird.</p> <p>In KOP darf die Anweisung N_TRIG nicht am Anfang oder Ende eines Netzwerks angeordnet werden. In FUP darf die Anweisung N_TRIG an einer beliebigen Stelle angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p>

1 In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 7- 16 Datentypen für die Parameter (Kontakte/Spulen P und N, P=, N=, P\_TRIG und N\_TRIG)

Parameter	Datentyp	Beschreibung
M_BIT	Bool	Merker, in dem der vorherige Zustand des Eingangs gespeichert wird
IN	Bool	Eingangsbit, dessen Flanke zu erkennen ist
OUT	Bool	Ausgangsbit, das angibt, dass eine Flanke erkannt wurde
CLK	Bool	Signalfluss oder Eingangsbit, dessen Flanke zu erkennen ist
Q	Bool	Ausgang, der angibt, dass eine Flanke erkannt wurde

Alle Flankenoperationen nutzen einen Merker (M\_BIT), um den vorherigen Zustand des beobachteten Eingangssignals zu speichern. Eine Flanke wird durch Vergleichen des Zustands des Eingangs mit dem Zustand des Merkers erkannt. Wenn die Zustände am Eingang auf einen Signalwechsel in der gewünschten Richtung hinweisen, wird eine Flanke gemeldet, indem der Ausgang auf WAHR gesetzt wird. Andernfalls wird der Ausgang auf FALSCH gesetzt.

**Hinweis**

Flankenoperationen werten den Eingang und die Merkerwerte bei jeder Ausführung aus, auch bei der ersten Ausführung. Sie müssen die Ausgangszustände des Eingangs und des Merkers in Ihrem Programm berücksichtigen, um die Flankenerkennung im ersten Zyklus zuzulassen oder nicht.

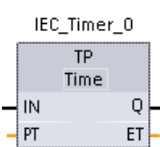
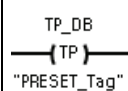
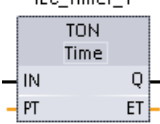
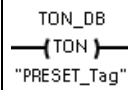
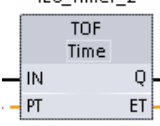
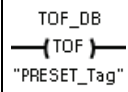
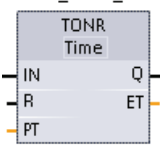
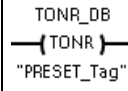
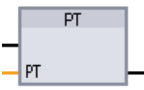
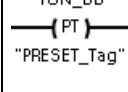
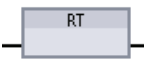

Weil der Merker von einer Ausführung zur nächsten gespeichert werden muss, müssen Sie für jede Flankenoperation ein eindeutiges Bit verwenden. Dieses Bit dürfen Sie nicht an anderen Stellen in Ihrem Programm nutzen. Vermeiden Sie außerdem temporären Speicher und Speicher, der von anderen Systemfunktionen geändert werden kann, z. B. E/A-Aktualisierungen. Verwenden Sie nur Merker (M), globale DBs oder statischen Speicher (in einem Instanz-DB) für M\_BIT-Speicherzuweisungen.



## 7.2 Zeiten

Mit den Zeitanweisungen können Sie programmierte Zeitverzögerungen einrichten. Die Anzahl der Zeiten, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Jede Zeit nutzt eine 16 Byte große DB-Struktur vom Datentyp IEC\_Timer, um Zeitdaten zu speichern, die im oberen Bereich der Box- oder Spulenanweisung angegeben werden. STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 7- 17 Zeitoperationen

KOP/FUP-Boxen	KOP-Spulen	SCL	Beschreibung
		<pre>"IEC_Timer_0_DB".TP (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	Die Zeit TP erzeugt einen Impuls mit einer voreingestellten Dauer.
		<pre>"IEC_Timer_0_DB".TON (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	Die Zeit TON setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN.
		<pre>"IEC_Timer_0_DB".TOF (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	Die Zeit TOF setzt den Ausgang Q nach einer voreingestellten Zeit auf AUS zurück.
		<pre>"IEC_Timer_0_DB".TONR (   IN:=_bool_in_,   R:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	Die Zeit TONR setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN. Die abgelaufene Zeit wird über mehrere Zeitintervalle kumuliert, bis Eingang R zum Zurücksetzen der abgelaufenen Zeit angestoßen wird.
Nur FUP: 		(Keine Entsprechung in SCL.)	Die Spule PT (Voreingestellte Zeit) lädt einen neuen PRESET-Zeitwert in den angegebenen IEC_Timer.
Nur FUP: 		(Keine Entsprechung in SCL.)	Die Spule RT (Zeit rücksetzen) setzt den angegebenen IEC_Timer zurück.

1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

2 In den SCL-Beispielen ist "IEC\_Timer\_0\_DB" der Name des Instanz-DBs.

Tabelle 7- 18 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Box: IN Spule: Signalfluss	Bool	TP, TON und TONR: Box: 0 = Zeit deaktivieren, 1 = Zeit aktivieren Spule: Kein Signalfluss = Zeit deaktivieren, Signalfluss = Zeit aktivieren TOF: Box: 0 = Zeit aktivieren, 1 = Zeit deaktivieren Spule: Kein Signalfluss = Zeit aktivieren, Signalfluss = Zeit deaktivieren
R	Bool	Nur TONR-Box: 0 = Nicht zurücksetzen 1= Abgelaufene Zeit und Q-Bit auf 0 zurücksetzen
Box: PT Spule: "PRESET_Tag"	Time	Zeitbox oder -spule: Eingang voreingestellte Zeit
Box: Q Spule: DBdata.Q	Bool	Zeitbox: Q-Boxausgang oder Q-Bit in den DB-Daten der Zeit Zeitspule: Sie können nur das Q-Bit in den DB-Daten der Zeit adressieren
Box: ET Spule: DBdata.ET	Time	Zeitbox: ET-Boxausgang (abgelaufene Zeit) oder ET-Zeitwert in den DB-Daten der Zeit Zeitspule: Sie können nur den ET-Zeitwert in den DB-Daten der Zeit adressieren

Tabelle 7- 19 Auswirkung von Wertänderungen in den Parametern PT und IN

Zeitschaltuhr	Änderungen der Box-Parameter PT und IN und der entsprechenden Spulenparameter
TP	<ul style="list-style-type: none"> <li>• Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen.</li> <li>• Ändert sich IN, während die Zeit läuft, hat dies keine Auswirkungen.</li> </ul>
TON	<ul style="list-style-type: none"> <li>• Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen.</li> <li>• Wenn IN nach FALSCH wechselt, während die Zeit läuft, wird die Zeit angehalten und zurückgesetzt.</li> </ul>
TOF	<ul style="list-style-type: none"> <li>• Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen.</li> <li>• Wenn IN nach WAHR wechselt, während die Zeit läuft, wird die Zeit angehalten und zurückgesetzt.</li> </ul>
TONR	<ul style="list-style-type: none"> <li>• Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen. Es hat dann Auswirkungen, wenn die Zeit fortgesetzt wird.</li> <li>• Wenn IN nach FALSCH wechselt, während die Zeit läuft, wird die Zeit angehalten, jedoch nicht zurückgesetzt. Wenn IN wieder nach WAHR wechselt, beginnt die Zeit ab dem kumulierten Zeitwert zu laufen.</li> </ul>

Die Werte von PT (voreingestellte Zeit) und ET (abgelaufene Zeit) werden in den Daten des angegebenen DBs IEC\_TIMER als vorzeichenbehaftete doppelte Ganzzahlen gespeichert, die einen Zeitwert in Millisekunden darstellen. Der Datentyp TIME verwendet die Kennung T# und kann als einfache Zeiteinheit (T#200ms oder 200) oder als zusammengesetzte Zeiteinheiten wie T#2s\_200ms eingegeben werden.

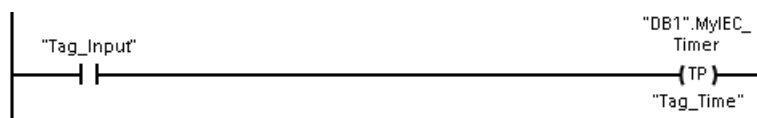
Tabelle 7- 20 Größe und Bereich des Datentyps TIME

Datentyp	Größe	Gültige Zahlenbereiche <sup>1</sup>
TIME	32 Bit, gespeichert als DInt -Daten	T#-24d_20h_31m_23s_648ms bis T#24d_20h_31m_23s_647ms Gespeichert als -2.147.483.648 ms bis +2.147.483.647 ms

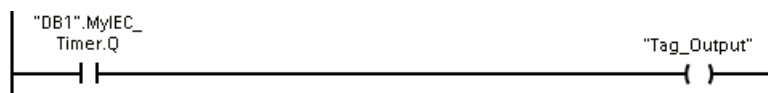
- <sup>1</sup> Der negative Bereich des oben dargestellten Datentyps TIME kann für die Zeiten nicht verwendet werden. Negative Werte für PT (voreingestellte Zeit) werden bei Ausführung der Zeitoperation auf Null gesetzt. ET (abgelaufene Zeit) ist immer ein positiver Wert.

### Beispiel für eine Zeitspule

Die Zeitspulen -(TP)-, -(TON)-, -(TOF)- und -(TONR)- müssen als letzte Anweisung in einem KOP-Netzwerk angeordnet werden. Wie im Beispiel für eine Zeit gezeigt, wertet eine Kontaktanweisung in einem nachfolgenden Netzwerk das Q-Bit in den Daten des DBs IEC\_Timer einer Zeitspule aus. Ebenso müssen Sie das Element ELAPSED in den Daten des DBs IEC\_Timer adressieren, wenn Sie den Wert der abgelaufenen Zeit in Ihrem Programm verwenden möchten.



Der Impulszeitgeber wird bei einem Wechsel von 0 nach 1 des Bitwerts von Tag\_Input gestartet. Die Zeit läuft für die vom Zeitwert Tag\_Time angegebene Zeitdauer.



Solange die Zeit läuft, ist der Zustand von DB1.MyIEC\_Timer.Q = 1 und von Tag\_Output = 1. Wenn der Wert von Tag\_Time abgelaufen ist, ist DB1.MyIEC\_Timer.Q = 0 und Tag\_Output = 0.

**Spulen Zeit rücksetzen -(RT)- und Zeitdauer laden -(PT)-**

Diese Spulenanweisungen können mit Box- oder Spulenzeiten verwendet werden und in der Mitte eines Strompfads angeordnet werden. Der Signalzustand des Spulenausgangs ist immer der gleiche wie am Spuleneingang. Wenn die Spule -(RT)- aktiviert wird, wird das Zeitelement ELAPSED in den Daten des angegebenen DBs IEC\_Timer auf 0 zurückgesetzt. Wenn die Spule -(PT)- aktiviert wird, wird das Zeitelement PRESET in den Daten des angegebenen DBs IEC\_Timer auf 0 zurückgesetzt.

**Hinweis**

Wenn Sie Zeiten in einem FB platzieren, können Sie die Option "Multiinstanz-Datenbaustein" auswählen. Die Namen der Zeitstrukturen können bei verschiedenen Datenstrukturen unterschiedlich sein, doch die Zeitdaten befinden sich in einem einzigen Datenbaustein, und es ist nicht für jede Zeit ein eigener Datenbaustein erforderlich. Dadurch verringert sich die Verarbeitungszeit und der für die Verwaltung der Zeiten erforderliche Datenspeicher. Zwischen den Datenstrukturen der Zeiten im gemeinsam genutzten Multiinstanz-DB gibt es keine Wechselwirkungen.

**Funktionsweise der Zeiten**

Tabelle 7- 21 Typen von IEC-Zeiten

Zeit	Zeitdiagramm
<p><b>TP: Impuls</b> Die Zeit TP erzeugt einen Impuls mit einer voreingestellten Dauer.</p>	
<p><b>TON: Einschaltverzögerung</b> Die Zeit TON setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN.</p>	

Zeit	Zeitdiagramm
<p><b>TOF:</b> Ausschaltverzögerung</p> <p>Die Zeit TON setzt den Ausgang Q nach einer voreingestellten Zeit auf AUS zurück.</p>	
<p><b>TONR:</b> Zeit akkumulieren</p> <p>Die Zeit TONR setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN. Die abgelaufene Zeit wird über mehrere Zeitintervalle kumuliert, bis Eingang R zum Zurücksetzen der abgelaufenen Zeit angestoßen wird.</p>	

### Hinweis

In der CPU ist keiner spezifischen Zeit eine bestimmte Ressource zugeordnet. Stattdessen nutzt jede Zeit eine eigene Zeitstruktur im DB-Speicher und eine fortlaufend in Betrieb befindliche interne CPU-Zeit, um die Zeitsteuerung durchzuführen.

Wenn eine Zeit aufgrund eines Flankenwechsels am Eingang einer Anweisung TP, TON, TOF oder TONR gestartet wird, wird der Wert der fortlaufend in Betrieb befindlichen internen CPU-Zeit in das START-Element der DB-Struktur kopiert, die dieser Zeit zugeordnet ist. Dieser Startwert bleibt solange unverändert, wie die Zeit läuft, und wird später bei jeder Aktualisierung der Zeit verwendet. Jedesmal, wenn die Zeit gestartet wird, wird ein neuer Startwert aus der internen CPU-Zeit in die Zeitstruktur geladen.

Wenn eine Zeit aktualisiert wird, wird der oben beschriebene Startwert vom aktuellen Wert der internen CPU-Zeit subtrahiert, um die abgelaufene Zeit zu ermitteln. Die abgelaufene Zeit wird dann mit der Voreinstellung verglichen, um den Zustand des Zeitbits Q zu ermitteln. Die Elemente ELAPSED und Q werden dann in der DB-Struktur, die dieser Zeit zugeordnet ist, aktualisiert. Beachten Sie, dass die abgelaufene Zeit an den voreingestellten Wert geknüpft ist (die Zeit akkumuliert die abgelaufene Zeit nicht weiter, wenn die Voreinstellung erreicht ist).

Die Zeit wird nur dann aktualisiert, wenn:

- Eine Zeitanweisung (TP, TON, TOF oder TONR) ausgeführt wird
- Das Element "ELAPSED" der Zeitstruktur im DB direkt von einer Anweisung referenziert wird
- Das Element "Q" der Zeitstruktur im DB direkt von einer Anweisung referenziert wird

## Programmierung von Zeiten

Die folgenden Konsequenzen von Zeiten sind bei der Planung und Erstellung Ihres Anwenderprogramms zu berücksichtigen:

- Sie können mehrere Aktualisierungen einer Zeit im gleichen Zyklus haben. Die Zeit wird bei jeder Ausführung der Zeitanweisung (TP, TON, TOF, TONR) aktualisiert sowie jedesmal, wenn das Element ELAPSED oder Q der Zeitstruktur als Parameter einer anderen ausgeführten Anweisung verwendet wird. Dies ist ein Vorteil, wenn Sie die neuesten Zeitdaten benötigen (praktisch ein direktes Auslesen der Zeit). Wenn Sie jedoch während eines Programmzyklus konsistente Werte nutzen möchten, ordnen Sie Ihre Zeitanweisung vor allen anderen Anweisungen, die diese Werte benötigen, an und verwenden statt der Elemente ELAPSED und Q der Zeitstruktur im DB die Variablen der Ausgänge Q und ET der Zeitanweisung.
- Sie können Zyklen ohne Zeitaktualisierung haben. Es ist möglich, Ihre Zeit in einer Funktion zu starten und diese Funktion dann für einen oder weitere Zyklen nicht mehr aufzurufen. Wenn keine anderen Anweisungen ausgeführt werden, die die Elemente ELAPSED oder Q der Zeitstruktur referenzieren, wird die Zeit nicht mehr aktualisiert. Eine neue Aktualisierung tritt erst auf, wenn entweder die Zeitanweisung erneut ausgeführt wird oder eine andere Anweisung ausgeführt wird, die ELAPSED oder Q aus der Zeitstruktur als Parameter nutzt.
- Es ist zwar nicht gerade üblich, aber Sie können mehreren Zeitanweisungen die gleiche DB-Zeitstruktur zuweisen. In Allgemeinen sollten Sie, um unerwartete Wechselwirkungen zu vermeiden, nur eine Zeit (TP, TON, TOF, TONR) pro DB-Zeitstruktur verwenden.
- Selbstrücksetzende Zeiten sind bei Trigger-Aktionen nützlich, die regelmäßig auftreten sollen. Typischerweise werden selbstrücksetzende Zeiten erstellt, indem ein Öffnerkontakt, der das Zeitbit referenziert, vor der Zeitanweisung angeordnet wird. Dieses Zeitnetzwerk befindet sich typischerweise oberhalb eines oder mehrerer abhängiger Netzwerke, die mit dem Zeitbit Aktionen auslösen. Wenn die Zeit abläuft (die abgelaufene Zeit erreicht den voreingestellten Wert), ist das Zeitbit einen Zyklus lang EIN, so dass die Logik der vom Zeitbit gesteuerten abhängigen Netzwerke ausgeführt werden kann. Bei der nächsten Ausführung des Zeitnetzwerks ist der Öffnerkontakt AUS, wodurch die Zeit zurückgesetzt und das Zeitbit gelöscht wird. Im nächsten Zyklus ist der Öffnerkontakt EIN, weshalb die Zeit neu gestartet wird. Beim Erstellen von selbstrücksetzenden Zeiten wie dieser verwenden Sie das Element "Q" der Zeitstruktur im DB nicht als den Parameter für den Öffnerkontakt vor der Zeitanweisung. Verwenden Sie hierfür stattdessen die an den Ausgang "Q" der Zeitanweisung angeschlossene Variable. Der Grund, weshalb vermieden werden sollte, auf das Element Q der DB-Zeitstruktur zuzugreifen, ist der, dass dadurch eine Aktualisierung der Zeit verursacht wird. Und wenn die Zeit durch den Öffnerkontakt aktualisiert wird, setzt der Kontakt die Zeitanweisung sofort zurück. Der Ausgang Q der Zeitanweisung ist während dieses einen Zyklus nicht EIN, und die abhängigen Netzwerke werden nicht ausgeführt.

## Speichern von Zeitdaten nach einem RUN-STOP-RUN-Wechsel oder einem Neustart der CPU

Wenn eine Session im Betriebszustand RUN mit dem Betriebszustand STOP oder einem Neustart der beendet wird und eine neue Session im Betriebszustand RUN gestartet wird, gehen die in der vorherigen RUN-Session gespeicherten Zeitdaten verloren, sofern die Zeitdatenstruktur nicht als remanent definiert ist (Zeiten TP, TON, TOF und TONR).

Wenn Sie beim Einfügen einer Zeitanweisung im Programmiereditor im Dialog der Aufrufoptionen die Standardeinstellung übernehmen, wird automatisch ein Instanz-DB zugewiesen, der **nicht als remanent definiert werden kann**. Um Ihre Zeitdaten als remanent zu definieren, müssen Sie entweder einen globalen DB oder einen Multiinstanz-DB nutzen.

## Zeitdaten durch Zuweisen eines globalen DB als remanente Daten speichern

Diese Option funktioniert unabhängig davon, wo die Zeit platziert wird (OB, FC oder FB).

### 1. Globalen DB erstellen:

- Doppelklicken Sie in der Projektnavigation auf "Neuen Baustein hinzufügen".
- Klicken Sie auf das Symbol für Datenbaustein (DB).
- Wählen Sie globalen DB als Typ.
- Wenn Sie einzelne Datenelemente im DB als remanent definieren möchten, müssen Sie für den DB-Typ das Kontrollkästchen "Optimiert" aktivieren. Die andere Option für den DB-Typ "Standard - kompatibel mit S7-300/400" gestattet es lediglich, entweder alle Datenelemente des DBs oder keine Datenelemente als remanent zu definieren.
- Klicken Sie auf "OK".

### 2. Zeitstruktur(en) zum DB hinzufügen:

- Fügen Sie im neuen globalen DB eine neue statische Variable vom Datentyp IEC\_Timer ein.
- Aktivieren Sie das Kontrollkästchen in der Spalte "Remanent", damit diese Struktur remanent ist.
- Wiederholen Sie diesen Vorgang, um für alle Zeiten, die Sie in diesem DB speichern möchten, Strukturen anzulegen. Sie können entweder jede Zeitstruktur in einem eindeutigen globalen DB anordnen, oder Sie können mehrere Zeitstrukturen in demselben globalen DB platzieren. Sie können in diesem globalen DB außer Zeiten auch andere statische Variablen anordnen. Durch Platzieren mehrerer Zeitstrukturen in demselben globalen DB können Sie die Gesamtzahl der Bausteine verringern.
- Benennen Sie die Zeitstrukturen ggf. um.

### 3. Öffnen Sie den Programmbaustein (OB, FC oder FB), in den Sie eine remanente Zeit einfügen möchten, für die Bearbeitung.

### 4. Platzieren Sie die Zeitanweisung an der gewünschten Stelle.

### 5. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf "Abbrechen".

### 6. Geben Sie oben in der neuen Zeitanweisung den Namen des globalen DB und der Zeitstruktur ein, die Sie oben erstellt haben (wählen Sie keinen Namen aus). Beispiel: "Data\_block\_3.Static\_1".

### Zeitdaten durch Zuweisen eines Multiinstanz-DB als remanente Daten speichern

Diese Möglichkeit funktioniert nur, wenn Sie die Zeit in einem FB ablegen.

Diese Variante ist davon abhängig, ob der FB mit "optimiertem" Bausteinzugriff angelegt wurde (nur symbolische Adressierung möglich). Nachdem der FB angelegt wurde, können Sie das Kontrollkästchen "Optimiert" nicht mehr ändern. Sie müssen die Option beim Anlegen des FB auswählen, und zwar im ersten Bild nach Auswahl des Befehl "Neuen Baustein hinzufügen" in der Projektnavigation. Um zu prüfen, wie das Zugriffsattribut eines vorhandenen FB konfiguriert ist, klicken Sie mit der rechten Maustaste in der Projektnavigation auf den FB, wählen "Eigenschaften" und dann "Attribute".

Wenn der FB mit aktiviertem Kontrollkästchen "Optimiert" angelegt wurde (nur symbolische Adressierung möglich), dann:

1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zeitanweisung an der gewünschten Stelle im FB.
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie der Zeit im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zeitanweisung erscheint im Editor und die IEC\_TIMER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Öffnen Sie ggf. den FB-Schnittstelleneditor (Sie müssen möglicherweise auf den kleinen Pfeil klicken, um die Ansicht zu vergrößern).
7. Suchen Sie unter "Statisch" die Zeitstruktur, die gerade für Sie angelegt wurde.
8. Ändern Sie in der Spalte "Remanent" dieser Zeitstruktur die Auswahl in "Remanent". Immer wenn dieser FB später in einem anderen Programmbaustein aufgerufen wird, wird ein Instanz-DB mit dieser Schnittstellendefinition angelegt, bei der die Zeitstruktur als remanent definiert ist.

Wenn der FB mit aktiviertem Kontrollkästchen "Standard - kompatibel mit S7-300/400" angelegt wurde (symbolische Adressierung und direkter Zugriff möglich), dann:

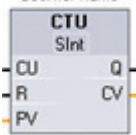
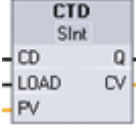
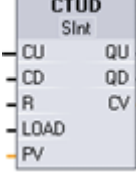
1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zeitanweisung an der gewünschten Stelle im FB.
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie der Zeit im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zeitanweisung erscheint im Editor und die IEC\_TIMER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Öffnen Sie den Baustein, der diesen FB verwenden soll.
7. Platzieren Sie diesen FB an der gewünschten Stelle. Dadurch wird ein Instanz-Datenbaustein für diesen FB angelegt.



8. Öffnen Sie den Instanz-Datenbaustein, der angelegt wurde, als Sie den FB im Editor platziert haben.
9. Suchen Sie unter "Statisch" die gewünschte Zeitstruktur. Aktivieren Sie in der Spalte "Remanent" dieser Zeitstruktur das Kontrollkästchen, um diese Struktur als remanent zu definieren.

## 7.3 Zähler

Tabelle 7- 22 Zähleranweisungen

KOP/FUP	SCL	Beschreibung
	<pre>"IEC_Counter_0_DB".CTU(   CU:=_bool_in,   R:=_bool_in,   PV:=_int_in,   Q=&gt;_bool_out,   CV=&gt;_int_out);</pre>	<p>Mit den Zähleranweisungen können Sie programminterne Ereignisse und externe Prozessereignisse zählen. Jeder Zähler nutzt eine in einem Datenbaustein abgelegte Struktur, um die Daten des Zählers zu speichern. Sie weisen den Datenbaustein zu, wenn Sie die Zähloperation im Editor einfügen.</p> <ul style="list-style-type: none"> <li>• CTU zählt vorwärts.</li> <li>• CTD zählt rückwärts.</li> <li>• CTUD zählt vorwärts und rückwärts.</li> </ul>
	<pre>"IEC_Counter_0_DB".CTD(   CD:=_bool_in,   LD:=_bool_in,   PV:=_int_in,   Q=&gt;_bool_out,   CV=&gt;_int_out);</pre>	
	<pre>"IEC_Counter_0_DB".CTUD(   CU:=_bool_in,   CD:=_bool_in,   R:=_bool_in,   LD:=_bool_in,   PV:=_int_in,   QU=&gt;_bool_out,   QD=&gt;_bool_out,   CV=&gt;_int_out);</pre>	

- 1 In KOP und FUP: Wählen Sie den Datentyp für den Zählwert aus der Klappliste unterhalb des Anweisungsnamens aus.
- 2 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 3 In den SCL-Beispielen ist "IEC\_Counter\_0\_DB" der Name des Instanz-DBs.

Tabelle 7- 23 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
CU, CD	Bool	Aufwärts- oder Abwärtszählen um jeweils eine Einheit
R (CTU, CTUD)	Bool	Zählwert auf Null zurücksetzen
LD (CTD, CTUD)	Bool	Ladesteuerung für den voreingestellten Wert
PV	SInt, Int, DInt, USInt, UInt, UInt	Voreingestellter Zählwert
Q, QU	Bool	Wahr, wenn CV >= PV

Parameter	Datentyp <sup>1</sup>	Beschreibung
QD	Bool	Wahr, wenn CV ≤ 0
CV	SInt, Int, DInt, USInt, UInt, UDInt	Aktueller Zählwert

<sup>1</sup> Der numerische Bereich der Zählwerte hängt vom ausgewählten Datentyp ab. Ist der Zählwert ein ganzzahliger Wert ohne Vorzeichen, können Sie bis Null herunter- oder bis zur Bereichsgrenze hochzählen. Ist der Zählwert ein ganzzahliger Wert mit Vorzeichen, können Sie bis zum unteren Grenzwert herunter- und bis zum oberen Grenzwert hochzählen.

Die Anzahl der Zähler, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Zähler benötigen den folgenden Speicherplatz:

- Bei den Datentypen SInt oder USInt benötigt die Zähleranweisung 3 Byte.
- Bei den Datentypen Int oder UInt benötigt die Zähleranweisung 6 Byte.
- Bei den Datentypen DInt oder UDInt benötigt die Zähleranweisung 12 Byte.

Diese Anweisungen nutzen Softwarezähler, deren maximale Zählgeschwindigkeit durch die Ausführungsrate des OBs, in den sie eingefügt wurden, begrenzt ist. Der OB, in den die Anweisungen eingefügt werden, muss häufig genug ausgeführt werden, um alle Transitionen der Eingänge CU oder CD zu erkennen. Um schneller zu zählen, nutzen Sie die Anweisung CTRL\_HSC (Seite 357).

**Hinweis**

Wenn Sie Zähleranweisungen in einem FB platzieren, können Sie die Option "Multiinstanz-DB" auswählen. Die Namen der Zählerstrukturen können bei verschiedenen Datenstrukturen unterschiedlich sein, doch die Zählerdaten befinden sich in einem einzigen DB, und es ist nicht für jeden Zähler ein eigener DB erforderlich. Dadurch verringert sich die Verarbeitungszeit und der für die Zähler erforderliche Datenspeicher. Zwischen den Datenstrukturen der Zähler im gemeinsam genutzten Multiinstanz-DB gibt es keine Wechselwirkungen.

**Funktionsweise der Zähler**

Tabelle 7- 24 Funktionsweise des Zählers CTU

Zähler	Bedienung
<p>Der Zähler CTU zählt um 1 vorwärts, wenn der Wert des Parameters CU von 0 nach 1 wechselt. Das CTU-Zeitdiagramm zeigt die Funktionsweise bei einem vorzeichenlosen ganzzahligen Zählwert (dabei ist PV = 3).</p> <ul style="list-style-type: none"> <li>• Ist der Wert des Parameters CV (aktueller Zählwert) größer oder gleich dem Wert des Parameters PV (voreingestellter Zählwert), dann lautet der Parameter für den Zählerausgang Q = 1.</li> <li>• Wenn der Wert des Rücksetzparameters R von 0 nach 1 wechselt, wird der aktuelle Zählwert auf 0 zurückgesetzt.</li> </ul>	<p>The diagram shows four signals over time: CU (counting input), R (reset input), CV (current count), and Q (output). CU is a periodic square wave. R is a single pulse. CV starts at 0 and increments by 1 each time CU transitions from 0 to 1. When R transitions from 0 to 1, CV resets to 0. When CV reaches the value 3 (PV) and CU is active, Q transitions from 0 to 1. The diagram is divided into four numbered intervals (1-4) by vertical dashed lines.</p>

Tabelle 7- 25 Funktionsweise des Zählers CTD

Zähler	Bedienung
<p>Der Zähler CTD zählt um 1 rückwärts, wenn der Wert des Parameters CD von 0 nach 1 wechselt. Das CTD-Zeitdiagramm zeigt die Funktionsweise bei einem vorzeichenlosen ganzzahligen Zählwert (dabei ist PV = 3).</p> <ul style="list-style-type: none"> <li>Ist der Wert des Parameters CV (aktueller Zählwert) kleiner oder gleich 0, so lautet der Parameter für den Zählerausgang Q = 1.</li> <li>Wechselt der Wert von Parameter LOAD von 0 nach 1, wird der Wert an Parameter PV (voreingestellter Wert) als neuer CV (aktueller Zählwert) in den Zähler geladen.</li> </ul>	

Tabelle 7- 26 Funktionsweise des Zählers CTUD

Zähler	Bedienung
<p>Der Zähler CTUD zählt um 1 vorwärts oder rückwärts, wenn der Vorwärtszähleingang (CU) oder der Rückwärtszähleingang (CD) von 0 nach 1 wechselt. Das CTUD-Zeitdiagramm zeigt die Funktionsweise bei ganzzahligem Zählwert ohne Vorzeichen (und PV = 4).</p> <ul style="list-style-type: none"> <li>Ist der Wert des Parameters CV größer oder gleich dem Wert des Parameters PV, dann lautet der Parameter für den Zählerausgang QU = 1.</li> <li>Ist der Wert des Parameters CV kleiner oder gleich 0, so lautet der Parameter für den Zählerausgang QD = 1.</li> <li>Wechselt der Wert von Parameter LOAD von 0 nach 1, wird der Wert an Parameter PV als neuer CV in den Zähler geladen.</li> <li>Wenn der Wert des Rücksetzparameters R von 0 nach 1 wechselt, wird der aktuelle Zählwert auf 0 zurückgesetzt.</li> </ul>	

### Speichern von Zählerdaten nach einem RUN-STOP-RUN-Wechsel oder einem Neustart der CPU

Wenn eine Session im Betriebszustand RUN mit dem Betriebszustand STOP oder einem Neustart der CPU beendet wird und eine neue Session im Betriebszustand RUN gestartet wird, gehen die in der vorherigen RUN-Session gespeicherten Zählerdaten verloren, sofern die Zählerdatenstruktur nicht als remanent definiert ist (Zeiten CTU, CTD und CTUD).

Wenn Sie beim Einfügen einer Zähleranweisung im Programmiereditor im Dialog der Aufrufoptionen die Standardeinstellung übernehmen, wird automatisch ein Instanz-DB zugewiesen, der **nicht als remanent definiert werden kann**. Um Ihre Zählerdaten als remanent zu definieren, müssen Sie entweder einen globalen DB oder einen Multiinstanz-DB nutzen.

### Zählerdaten durch Zuweisen eines globalen DBs als remanente Daten speichern

Diese Option funktioniert unabhängig davon, wo der Zähler platziert wird (OB, FC oder FB).

1. Globalen DB erstellen:
  - Doppelklicken Sie in der Projektnavigation auf "Neuen Baustein hinzufügen".
  - Klicken Sie auf das Symbol für Datenbaustein (DB).
  - Wählen Sie globalen DB als Typ.
  - Wenn Sie einzelne Elemente im DB als remanent definieren möchten, müssen Sie das Kontrollkästchen "Nur symbolisch adressierbar" aktivieren.
  - Klicken Sie auf "OK".
2. Zählerstruktur(en) zum DB hinzufügen:
  - Fügen Sie im neuen globalen DB eine neue statische Variable vom Datentyp Zähler ein. Berücksichtigen Sie dabei den Typ für die Voreinstellung und den Zählerwert.

Zählerdatentyp	Entsprechender Typ für die Voreinstellung und den Zählerwert
IEC_Counter	INT
IEC_SCounter	SINT
IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

1. Aktivieren Sie das Kontrollkästchen in der Spalte "Remanent", damit diese Struktur remanent ist.

- Wiederholen Sie diesen Vorgang, um für alle Zähler, die Sie in diesem DB speichern möchten, Strukturen anzulegen. Sie können entweder jede Zählerstruktur in einem eindeutigen globalen DB anordnen, oder Sie können mehrere Zählerstrukturen in demselben globalen DB platzieren. Sie können in diesem globalen DB außer Zählern auch andere statische Variablen anordnen. Durch Platzieren mehrerer Zählerstrukturen in demselben globalen DB können Sie die Gesamtzahl der Bausteine verringern.
  - Benennen Sie die Zählerstrukturen ggf. um.
2. Öffnen Sie den Programmbaustein (OB, FC oder FB), in den Sie einen remanenten Zähler einfügen möchten, für die Bearbeitung.
  3. Platzieren Sie die Zähleranweisung an der gewünschten Stelle.
  4. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf "Abbrechen". Nun sollte die neue Zähleranweisung mit dem Platzhalter "???" oberhalb und unterhalb des Anweisungsnamens angezeigt werden.
  5. Geben Sie oben in der neuen Zähleranweisung den Namen des globalen DBs und der Zählerstruktur ein, die Sie oben erstellt haben (wählen Sie keinen Namen aus). Beispiel: "Data\_block\_3.Static\_1". Daraufhin wird der entsprechende Typ für die Voreinstellung und den Zählerwert vorgegeben (Beispiel: UInt für eine IEC\_UCounter-Struktur).

### Zählerdaten durch Zuweisen eines Multiinstanz-DB als remanente Daten speichern

Diese Möglichkeit funktioniert nur, wenn Sie den Zähler in einem FB ablegen.

Diese Variante ist davon abhängig, ob der FB nur für symbolische Adressierung angelegt wurde. Nachdem der FB angelegt wurde, können Sie das Kontrollkästchen "Nur symbolisch adressierbar" nicht mehr ändern. Sie müssen die Option beim Anlegen des FB auswählen, und zwar im ersten Bild nach Auswahl des Befehl "Neuen Baustein hinzufügen" in der Projektnavigation. Um zu prüfen, wie ein vorhandener FB hinsichtlich dieser Eigenschaft konfiguriert ist, klicken Sie mit der rechten Maustaste in der Projektnavigation auf den FB, wählen "Eigenschaften" und dann "Attribute".

Wenn der FB mit aktiviertem Kontrollkästchen "Nur symbolisch adressierbar" angelegt wurde, dann:

1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zähleranweisung an der gewünschten Stelle im FB.
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie dem Zähler im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zähleranweisung erscheint im Editor mit dem Typ INT für die Voreinstellung und den Zählwert, und die IEC\_COUNTER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Ändern Sie ggf. den Typ in der Zähleranweisung von INT in einen anderen Typ. Die Zählerstruktur wird entsprechend verändert.

Typ in Zähleranweisung (für Voreinstellung und Zählwert)	Entsprechender Struktur-Typ in FB-Schnittstelle
INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

1. Öffnen Sie ggf. den FB-Schnittstelleneditor (Sie müssen möglicherweise auf den kleinen Pfeil klicken, um die Ansicht zu vergrößern).
2. Suchen Sie unter "Statisch" die Zählerstruktur, die gerade für Sie angelegt wurde.
3. Ändern Sie in der Spalte "Remanent" dieser Zählerstruktur die Auswahl in "Remanent". Immer wenn dieser FB später in einem anderen Programmbaustein aufgerufen wird, wird ein Instanz-DB mit dieser Schnittstellendefinition angelegt, bei der die Zählerstruktur als remanent definiert ist.

Wenn der FB mit *nicht* aktiviertem Kontrollkästchen "Nur symbolisch adressierbar" angelegt wurde, dann:

1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zähleranweisung an der gewünschten Stelle im FB.
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie dem Zähler im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zähleranweisung erscheint im Editor mit dem Typ INT für die Voreinstellung und den Zählwert, und die IEC\_COUNTER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Ändern Sie ggf. den Typ in der Zähleranweisung von INT in einen anderen Typ. Die Zählerstruktur wird entsprechend verändert.

Typ in Zähleranweisung (für Voreinstellung und Zählwert)	Entsprechender Struktur-Typ in FB-Schnittstelle
INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

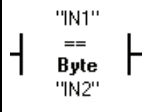
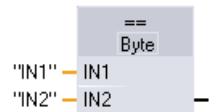
1. Öffnen Sie den Baustein, der diesen FB verwenden soll.
2. Platzieren Sie diesen FB an der gewünschten Stelle. Dadurch wird ein Instanz-Datenbaustein für diesen FB angelegt.

3. Öffnen Sie den Instanz-Datenbaustein, der angelegt wurde, als Sie den FB im Editor platziert haben.
4. Suchen Sie unter "Statisch" die gewünschte Zählerstruktur. Aktivieren Sie in der Spalte "Remanent" dieser Zählerstruktur das Kontrollkästchen, um diese Struktur als remanent zu definieren.

## 7.4 Vergleichen

### 7.4.1 Vergleichen

Tabelle 7- 27 Vergleichsoperationen

KOP	FUP	SCL	Beschreibung
		<pre> out := in1 = in2; or IF in1 = in2   THEN out := 1;   ELSE out := 0; END IF; </pre>	Vergleicht zwei Werte desselben Datentyps. Hat der KOP-Kontaktvergleich das Ergebnis WAHR, wird der Kontakt aktiviert. Ist das Ergebnis des FUP-Boxvergleichs WAHR, dann ist der Box-Ausgang ebenfalls WAHR.

- <sup>1</sup> In KOP und FUP: Klicken Sie auf den Anweisungsnamen (wie "=="), um den Vergleichstyp über die Klappliste zu ändern. Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 28 Datentypen für die Parameter

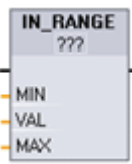

Parameter	Datentyp	Beschreibung
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, String, Char, Time, DTL, Konstante	Zu vergleichende Werte

Tabelle 7- 29 Beschreibungen der Vergleiche

Beziehungstyp	Der Vergleich ist wahr, wenn...
=	IN1 gleich IN2 ist
<>	IN1 nicht gleich IN2 ist
>=	IN1 größer oder gleich IN2 ist
<=	IN1 kleiner oder gleich IN2 ist
>	IN1 größer als IN2 ist
<	IN1 kleiner als IN2 ist

### 7.4.2 Operationen IN\_RANGE und OUT\_RANGE

Tabelle 7- 30 Anweisungen IN\_RANGE und OUT\_RANGE

KOP/FUP	SCL	Beschreibung
	<pre>out := IN_RANGE(min, val, max);</pre>	<p>Prüft, ob ein Eingabewert innerhalb oder außerhalb eines angegebenen Wertebereichs liegt.</p> <p>Ist das Ergebnis des Vergleichs WAHR, dann ist der Box-Ausgang ebenfalls WAHR.</p>
	<pre>out := OUT_RANGE(min, val, max);</pre>	

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 31 Datentypen für die Parameter

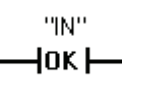

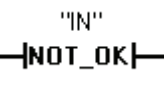

Parameter	Datentyp <sup>1</sup>	Beschreibung
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Konstante	Vergleichereingänge

<sup>1</sup> Die Eingangsparameter MIN, VAL und MAX müssen denselben Datentyp haben.

- Der Vergleich IN\_RANGE ist wahr, wenn: MIN <= VAL <= MAX
- Der Vergleich OUT\_RANGE ist wahr, wenn: VAL < MIN oder VAL > MAX

### 7.4.3 Operationen OK und NOT\_OK

Tabelle 7- 32 Anweisungen OK und NOT\_OK

KOP	FUP	SCL	Beschreibung
		Nicht verfügbar	<p>Prüft, ob eine Eingabedatenreferenz eine gültige Realzahl nach der IEEE-Spezifikation 754 ist.</p>
		Nicht verfügbar	

<sup>1</sup> In KOP und FUP: Ist der KOP-Kontakt WAHR, wird der Kontakt aktiviert und leitet Signalfloss. Ist die FUP-Box WAHR, dann ist der Box-Ausgang WAHR.



Tabelle 7- 33 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Real, LReal	Eingangsdaten

Tabelle 7- 34 Funktionsweise

Anweisung	Die Prüfung auf eine reale Zahl ist WAHR, wenn:
OK	Der Eingabewert ist eine gültige Realzahl <sup>1</sup> .
NOT_OK	Der Eingabewert ist keine gültige Realzahl <sup>1</sup> .

<sup>1</sup> Ein Wert Real oder LReal ist ungültig, wenn er +/- INF (unendlich), NaN (Not a Number, keine Zahl) oder ein denormalisierter Wert ist. Ein denormalisierter Wert ist eine Zahl sehr nah bei Null. In Berechnungen ersetzt die CPU einen denormalisierten Wert durch Null.

### Siehe auch

EN und ENO in KOP, FUP und SCL (Seite 174)

## 7.5 Arithmetik

### 7.5.1 Anweisung Berechnen

Tabelle 7- 35 Anweisung CALCULATE

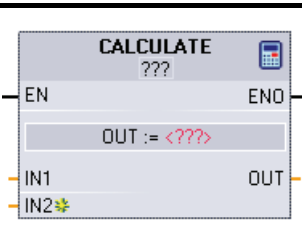
KOP/FUP	SCL	Beschreibung
	<p>Verwenden Sie die herkömmlichen mathematischen SCL-Ausdrücke, um die Gleichung zu erstellen.</p>	<p>Mit der Anweisung CALCULATE können Sie eine mathematische Funktion erstellen, die Eingänge (IN1, IN2, .. INn) verarbeitet und das Ergebnis an OUT entsprechend der von Ihnen vorgegebenen Gleichung ausgibt.</p> <ul style="list-style-type: none"> <li>Wählen Sie zunächst einen Datentyp aus. Alle Eingänge und der Ausgang müssen denselben Datentyp haben.</li> <li>Um einen weiteren Eingang hinzuzufügen, klicken Sie auf das Symbol am letzten Eingang.</li> </ul>

Tabelle 7- 36 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord

<sup>1</sup> Die Parameter IN und OUT müssen denselben Datentyp haben (mit impliziten Umwandlungen der Eingangsparameter). Beispiel: Ein Wert vom Typ SINT für einen Eingang würde in einen Wert vom Typ INT oder REAL umgewandelt werden, wenn OUT vom Typ INT oder REAL ist.

Klicken Sie auf das Taschenrechnersymbol, um den Dialog aufzurufen und ihre mathematische Funktion zu definieren. Sie geben Ihre Gleichung als Eingänge (wie IN1 und IN2) und Operationen ein. Wenn Sie auf "OK" klicken, um die Funktion zu speichern, erstellt der Dialog automatisch die Eingänge für die Anweisung CALCULATE.

Ein Beispiel und eine Liste der möglichen mathematischen Operationen, die Sie verwenden können, wird im unteren Bereich des Editors gezeigt.



**Hinweis**

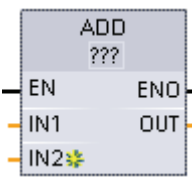
Sie müssen außerdem einen Eingang für die Konstanten in Ihrer Funktion anlegen. Der konstante Wert wird dann in den zugewiesenen Eingang der Anweisung CALCULATE eingegeben.

Indem Sie Konstanten als Eingänge eingeben, können Sie die Anweisung CALCULATE an andere Stellen in Ihrem Anwenderprogramm kopieren, ohne die Funktion ändern zu müssen. Sie können dann die Werte oder Variablen der Eingänge für die Anweisung ändern, ohne die Funktion zu verändern.

Wenn die Anweisung CALCULATE ausgeführt wird und alle einzelnen Anweisungen in der Berechnung erfolgreich durchgeführt werden, dann ist ENO = 1. Andernfalls ist ENO = 0.

## 7.5.2 Operationen Addieren, Subtrahieren, Multiplizieren und Dividieren

Tabelle 7- 37 Anweisungen Addieren, Subtrahieren, Multiplizieren und Dividieren

KOP/FUP	SCL	Beschreibung
	<pre> out := in1 + in2; out := in1 - in2; out := in1 * in2; out := in1 / in2; </pre>	<ul style="list-style-type: none"> <li>• ADD: Addieren (<math>IN1 + IN2 = OUT</math>)</li> <li>• SUB: Subtrahieren (<math>IN1 - IN2 = OUT</math>)</li> <li>• MUL: Multiplizieren (<math>IN1 * IN2 = OUT</math>)</li> <li>• DIV: Dividieren (<math>IN1 / IN2 = OUT</math>)</li> </ul> <p>Bei einer ganzzahligen Division werden die Nachkommastellen des Quotienten so verkürzt, dass ein ganzzahliger Ausgangswert entsteht.</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 38 Datentypen für die Parameter (KOP und FUP)

Parameter	Datentyp <sup>1</sup>	Beschreibung
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Konstante	Eingänge der arithmetischen Operation
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Ausgang der arithmetischen Operation

<sup>1</sup> Die Parameter IN1, IN2 und OUT müssen denselben Datentyp haben.



Um einen Eingang ADD oder MUL hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".

Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

Die arithmetische Anweisung führt, wenn sie aktiviert ist ( $EN = 1$ ), die angegebene Funktion für die Eingangswerte (IN1 und IN2) aus und speichert das Ergebnis in der vom Ausgangsparameter (OUT) angegebenen Speicheradresse. Nachdem die Operation erfolgreich ausgeführt ist, wird  $ENO = 1$  gesetzt.

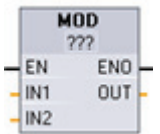
Tabelle 7- 39 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert der arithmetischen Operation liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Der niederwertigste Teil des Ergebnisses, der in die Zielgröße passt, wird zurückgegeben.
0	Division durch 0 ( $IN2 = 0$ ): Das Ergebnis ist undefiniert und Null wird zurückgegeben.
0	Real/LReal: Ist einer der Werte NaN (not a number, keine Zahl), so wird der Wert NaN zurückgegeben

ENO	Beschreibung
0	ADD Real/LReal: Sind beide Eingangswerte INF (unendlich) mit unterschiedlichen Vorzeichen, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.
0	SUB Real/LReal: Sind beide Eingangswerte INF (unendlich) mit demselben Vorzeichen, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.
0	MUL Real/LReal: Ist ein Eingangswert Null und der andere INF, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.
0	DIV Real/LReal: Sind beide Eingangswerte Null oder INF, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.

### 7.5.3 Modulo-Anweisung

Tabelle 7- 40 Anweisung MOD (Divisionsrest gewinnen)

KOP/FUP	SCL	Beschreibung
	<pre>out := in1 MOD in2;</pre>	<p>Mit der Anweisung MOD können Sie den Divisionsrest einer Ganzzahldivision ausgeben. Der Wert am Eingang IN1 wird durch den Wert am Eingang IN2 dividiert, und der Divisionsrest wird am Ausgang OUT ausgegeben.</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 41 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
IN1 und IN2	SInt, Int, DInt, USInt, UInt, UDInt, Konstante	Modulo-Eingänge
OUT	SInt, Int, DInt, USInt, UInt, UDInt	Modulo-Ausgang

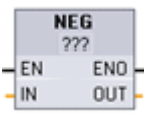
<sup>1</sup> Die Parameter IN1, IN2 und OUT müssen denselben Datentyp haben.

Tabelle 7- 42 ENO-Werte

ENO	Beschreibung
1	Kein Fehler
0	Wert IN2 = 0, OUT wird der Wert Null zugewiesen

## 7.5.4 Anweisung Negation

Tabelle 7- 43 Anweisung NEG

KOP/FUP	SCL	Beschreibung
	<code>-(in);</code>	Mit der Anweisung NEG wird das arithmetische Vorzeichen des Werts von Parameter IN umgekehrt und das Ergebnis im Parameter OUT gespeichert.

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 44 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
IN	SInt, Int, DInt, Real, LReal, Konstante	Eingang der arithmetischen Operation
OUT	SInt, Int, DInt, Real, LReal	Ausgang der arithmetischen Operation


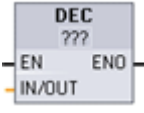
<sup>1</sup> Die Parameter IN und OUT müssen denselben Datentyp haben.

Tabelle 7- 45 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Beispiel für SInt: NEG (-128) resultiert in +128, was das Maximum für den Datentyp überschreitet.

## 7.5.5 Operationen INC (Inkrementieren) und DEC (Dekrementieren)

Tabelle 7- 46 Anweisungen INC und DEC

KOP/FUP	SCL	Beschreibung
	<code>in_out := in_out + 1;</code>	Erhöht einen ganzzahligen Wert mit oder ohne Vorzeichen: IN_OUT -Wert +1 = IN_OUT -Wert
	<code>in_out := in_out - 1;</code>	Verringert einen ganzzahligen Wert mit oder ohne Vorzeichen: IN_OUT -Wert - 1 = IN_OUT -Wert

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 47 Datentypen für die Parameter

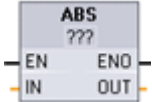
Parameter	Datentyp	Beschreibung
IN/OUT	SInt, Int, DInt, USInt, UInt, UDInt	Ein- und Ausgang der arithmetischen Operation

Tabelle 7- 48 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Beispiel für SInt: INC (+127) ergibt +128, was das Maximum für den Datentyp überschreitet.

## 7.5.6 Operation ABS (Absolutwert bilden)

Tabelle 7- 49 Anweisung ABS

KOP/FUP	SCL	Beschreibung
	<pre>out := ABS(in);</pre>	Berechnet den Absolutwert einer vorzeichenbehafteten Ganzzahl oder Realzahl an Parameter IN und speichert das Ergebnis in Parameter OUT.

- 1 In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 50 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
IN	SInt, Int, DInt, Real, LReal	Eingang der arithmetischen Operation
OUT	SInt, Int, DInt, Real, LReal	Ausgang der arithmetischen Operation

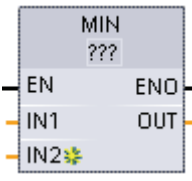
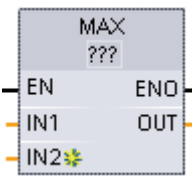
- 1 Die Parameter IN und OUT müssen denselben Datentyp haben.

Tabelle 7- 51 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert der arithmetischen Operation liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Beispiel für SInt: ABS (-128) resultiert in +128, was das Maximum für den Datentyp überschreitet.

## 7.5.7 Anweisungen MIN und MAX

Tabelle 7- 52 Anweisungen MIN und MAX

KOP/FUP	SCL	Beschreibung
	<pre>out:= MIN(     in1:=_variant_in_,     in2:=_variant_in_     [,...in32]);</pre>	Die Anweisung MIN vergleicht den Wert zweier Parameter IN1 und IN2 und weist den kleineren Wert dem Parameter OUT zu.
	<pre>out:= MAX(     in1:=_variant_in_,     in2:=_variant_in_     [,...in32]);</pre>	Die Anweisung MAX vergleicht den Wert zweier Parameter IN1 und IN2 und weist den größeren Wert dem Parameter OUT zu.

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 53 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
IN1, IN2 [...IN32]	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Konstante	Eingänge der arithmetischen Operation (max. 32 Eingänge)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Ausgang der arithmetischen Operation

<sup>1</sup> Die Parameter IN1, IN2 und OUT müssen denselben Datentyp haben.



Um einen Eingang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".


Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 7- 54 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Nur beim Datentyp Real: <ul style="list-style-type: none"> <li>• Einer oder beide Eingänge sind keine Realzahl (NaN).</li> <li>• Der resultierende Ausgang OUT ist +/- INF (unendlich).</li> </ul>

### 7.5.8 Anweisung LIMIT

Tabelle 7- 55 Anweisung LIMIT (Grenzwert setzen)

KOP/FUP	SCL	Beschreibung
	<pre>LIMIT (MN:=_variant_in_,         IN:=_variant_in_,         MX:=_variant_in_,         OUT:=_variant_out_);</pre>	<p>Mit der Anweisung Limit können Sie prüfen, ob der Wert von Parameter IN innerhalb des mit Parameter MIN und MAX and if not, clamps the value at MIN or MAX. vorgegebenen Wertebereichs liegt.</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 56 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
MN, IN und MX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Konstante	Eingänge der arithmetischen Operation
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Ausgang der arithmetischen Operation

<sup>1</sup> Die Parameter MN, IN, MX und OUT müssen denselben Datentyp haben.

Liegt der Wert von Parameter IN innerhalb des angegebenen Bereichs, so wird der Wert für IN in Parameter OUT gespeichert. Liegt der Wert von Parameter IN außerhalb des angegebenen Bereichs, wird in OUT der Wert von Parameter MIN (wenn der Wert IN kleiner als der Wert MIN ist) oder der Wert von Parameter MAX (wenn der Wert IN größer als der Wert MAX ist) ausgegeben.

Tabelle 7- 57 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Real: Ist einer der Werte für MIN, IN oder MAX keine Zahl (NaN), wird NaN zurückgegeben.
0	Ist MIN größer als MAX, wird der Wert IN dem Ausgang OUT zugewiesen.

SCL-Beispiele:

- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //Ergebnis: MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); //Ergebnis: MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); //Ergebnis: MyVal = 10

### 7.5.9 Arithmetische Gleitpunktoperationen

Mit den Gleitpunktanweisungen können Sie arithmetische Funktionen mit dem Datentyp Real oder LReal programmieren:

- SQR: Quadrat (IN <sup>2</sup> = OUT)
- SQRT: Quadratwurzel (√IN = OUT)





- LN: Natürlicher Logarithmus (LN(IN) = OUT)
- EXP: Natürliche Exponentialfunktion ( $e^{IN} = OUT$ ), mit der Basis  $e = 2.71828182845904523536$
- EXPT: Potenzieren ( $IN1^{IN2} = OUT$ )

Die EXPT-Parameter IN1 und OUT haben immer den gleichen Datentyp, für den Sie Real oder LReal auswählen müssen. Für den Exponentialparameter IN2 können Sie den Datentyp aus vielen Datentypen wählen.

- FRAC: Nachkommastellen (Nachkommastelle der Gleitpunktzahl IN = OUT)
- SIN: Sinus ( $\sin(IN \text{ radians}) = OUT$ )  
ASIN: Arcussinus ( $\arcsin(IN) = OUT \text{ radians}$ ), wobei  $\sin(OUT \text{ radians}) = IN$
- COS: Cosinus ( $\cos(IN \text{ radians}) = OUT$ )  
ACOS: Arcuscosinus ( $\arcsin(IN) = OUT \text{ radians}$ ), wobei  $\cos(OUT \text{ radians}) = IN$
- TAN: Tangens ( $\tan(IN \text{ radians}) = OUT$ )  
ATAN: Arcustangens ( $\arctan(IN) = OUT \text{ radians}$ ), wobei  $\tan(OUT \text{ radians}) = IN$

Tabelle 7- 58 Beispiele für arithmetische Gleitpunktanweisungen

KOP/FUP	SCL	Beschreibung
	<pre>out := SQR(in);</pre> oder <pre>out := in * in;</pre>	Quadrat: $IN^2 = OUT$ Beispiel: Wenn IN = 9, dann OUT = 81.
	<pre>out := in1 ** in2;</pre>	Potenzieren: $IN1^{IN2} = OUT$ Beispiel: Wenn IN1 = 3 und IN2 = 2, dann OUT = 9.

- <sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" (beim Anweisungsnamen) und wählen Sie den Datentyp in der Klappliste aus.
- <sup>2</sup> In SCL: Sie können die mathematischen Ausdrücke in SCL auch mit den grundlegenden mathematischen Operatoren erstellen.

Tabelle 7- 59 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN, IN1	Real, LReal, Konstante	Eingänge
IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Konstante	Eingang EXPT-Exponent
OUT	Real, LReal	Ausgänge

Tabelle 7- 60 ENO-Status

ENO	Anweisung	Bedingung	Ergebnis (OUT)
1	Alle	Kein Fehler	Gültiges Ergebnis
0	SQR	Ergebnis überschreitet den gültigen Bereich für Real/LReal	+INF
		IN ist +/- NaN (keine Zahl)	+NaN
	SQRT	IN ist negativ	-NaN
		IN ist +/- INF (unendlich) oder +/- NaN	+/- INF oder +/- NaN
	LN	IN ist 0,0, negativ, -INF oder -NaN	-NaN
		IN ist +INF oder +NaN	+INF oder +NaN
	EXP	Ergebnis überschreitet den gültigen Bereich für Real/LReal	+INF
		IN ist +/- NaN	+/- NaN
	SIN, COS, TAN	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN
	ASIN, ACOS	IN ist außerhalb des gültigen Bereichs von -1,0 bis +1,0	+NaN
		IN ist +/- NaN	+/- NaN
	ATAN	IN ist +/- NaN	+/- NaN
	FRAC	IN ist +/- INF oder +/- NaN	+NaN
	EXPT	IN1 ist +INF und IN2 ist nicht -INF	+INF
		IN1 ist negativ oder -INF	+NaN, wenn IN2 = Real/LReal, ansonsten -INF
		IN1 oder IN2 ist +/- NaN	+NaN
IN1 ist 0,0 und IN2 ist Real/LReal (nur)		+NaN	

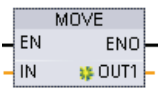
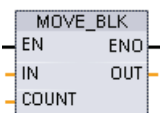
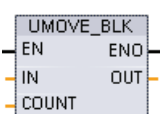
## 7.6 MOVE

### 7.6.1 Operationen MOV (Übertragen) und MOVE\_BLK (Wertebereich übertragen)

Mit den Übertragungsanweisungen kopieren Sie Datenelemente in eine neue Adresse im Speicher und wandeln die Daten von einem Datentyp in einen anderen um. Die Quelldaten werden dadurch nicht verändert.

- Mit der Anweisung MOVE wird ein einzelnes Datenelement von der mit Parameter IN angegebenen Quelladresse in die mit Parameter OUT angegebenen Zieladressen kopiert.
- Die Anweisungen MOVE\_BLK und UMOVE\_BLK verfügen zusätzlich über einen Parameter COUNT. Mit COUNT wird festgelegt, wie viele Datenelemente kopiert werden sollen. Die Anzahl der Bytes pro kopiertem Element hängt davon ab, welcher Datentyp den Variablennamen der Parameter IN und OUT in der PLC-Variablen-tabelle zugewiesen ist.

Tabelle 7- 61 Anweisungen MOVE, MOVE\_BLK und UMOVE\_BLK

KOP/FUP	SCL	Beschreibung
	<code>out1 := in;</code>	Kopiert ein unter einer bestimmten Adresse gespeichertes Datenelement in eine neue Adresse oder in mehrere Adressen. <sup>1</sup>
	<code>MOVE_BLK (   in:=_variant_in,   count:=_uint_in,   out=&gt;_variant_out);</code>	Unterbrechbare Übertragung, die einen Bereich mit Datenelementen in eine neue Adresse kopiert.
	<code>UMOVE_BLK (   in:=_variant_in,   count:=_uint_in,   out=&gt;_variant_out);</code>	Ununterbrechbare Übertragung, die einen Bereich mit Datenelementen in eine neue Adresse kopiert.

- <sup>1</sup> Anweisung MOVE: Um einen weiteren Ausgang in KOP oder FUP hinzuzufügen, klicken Sie auf das Symbol "Erstellen" neben dem Ausgangsparameter. In SCL verwenden Sie mehrere Zuweisungsanweisungen. Sie können auch eine der Schleifenkonstruktionen verwenden.

Tabelle 7- 62 Datentypen für die Anweisung MOVE

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	Quelladresse
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	Zieladresse



Um Ausgänge MOVE hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss und wählen den Befehl "Ausgang einfügen".

Um einen Ausgang zu löschen, klicken Sie bei einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss (sofern mehr als die zwei ursprünglichen Ausgänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 7- 63 Datentypen für die Anweisungen MOVE\_BLK und UMOVE\_BLK

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	Anfangsadresse der Quelle
COUNT	UInt	Anzahl der zu kopierenden Datenelemente
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	Anfangsadresse des Ziels

**Hinweis**

**Regeln für die Anweisungen zum Übertragen von Daten**

- Um Daten vom Datentyp Bool zu kopieren, verwenden Sie SET\_BF, RESET\_BF, R, S oder eine Ausgangsspule (KOP) (Seite 188).
- Um Daten eines einzelnen elementaren Datentyps zu kopieren, verwenden Sie MOVE.
- Um ein Feld eines elementaren Datentyps zu kopieren, verwenden Sie MOVE\_BLK oder UMOVE\_BLK.
- Um eine Struktur zu kopieren, verwenden Sie MOVE.
- Um eine Zeichenkette zu kopieren, verwenden Sie S\_MOVE (Seite 269).
- Um ein einzelnes Zeichen in einer Zeichenkette zu kopieren, verwenden Sie MOVE.
- Die Anweisungen MOVE\_BLK und UMOVE\_BLK können Sie nicht verwenden, um Felder oder Strukturen in die Speicherbereiche E, A oder M zu kopieren.

Die Anweisungen MOVE\_BLK und UMOVE\_BLK unterscheiden sich in der Verarbeitung von Alarmen:

- Alarmereignisse werden **in die Warteschlange gestellt** und während der Ausführung von MOVE\_BLK verarbeitet. Die Anweisung MOVE\_BLK nutzen Sie, wenn die Daten an der Zieladresse der Übertragung nicht in einem Unterprogramm eines Alarm-OBs verwendet werden, oder, sofern sie verwendet werden, die Zieldaten nicht konsistent sein müssen. Wenn eine Anweisung MOVE\_BLK unterbrochen wird, dann ist das zuletzt übertragene Datenelement an der Zieladresse vollständig und konsistent. Die Anweisung MOVE\_BLK wird nach Ausführung des Alarm-OBs fortgesetzt.
- Alarmereignisse werden **in die Warteschlange gestellt, aber erst verarbeitet**, wenn die Ausführung von UMOVE\_BLK beendet ist. Die Anweisung UMOVE\_BLK nutzen Sie, wenn die Übertragungsanweisung beendet und die Zieldaten konsistent sein müssen, bevor das Unterprogramm eines Alarm-OBs ausgeführt wird. Beachten Sie für weitere Informationen den Abschnitt zur Datenkonsistenz (Seite 163).

ENO ist nach der Ausführung der Anweisung MOVE immer wahr.

Tabelle 7- 64 ENO-Status

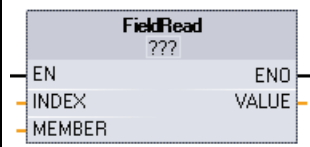
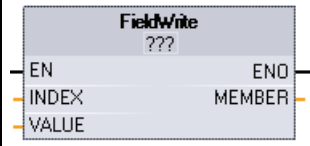
ENO	Bedingung	Ergebnis
1	Kein Fehler	Alle COUNT-Elemente wurden erfolgreich kopiert.
0	Der Quellbereich (IN) oder der Zielbereich (OUT) überschreitet den verfügbaren Speicherbereich.	Elemente, die passen, werden kopiert. Es werden keine Teilelemente kopiert.

## 7.6.2 Anweisungen FeldLesen und FeldSchreiben

### Hinweis

In STEP 7 V10.5 wurden eine Variablenreferenz als Array-Index oder multidimensionale Arrays **nicht unterstützt**. Mit den Anweisungen FeldLesen und FeldSchreiben wurden variable Array-Index-Anweisungen für ein eindimensionales Array bereitgestellt. In STEP 7 V11 **werden** eine Variablenreferenz als Array-Index und multidimensionale Arrays **unterstützt**. FeldLesen und FeldSchreiben sind in STEP 7 V11 enthalten und sind abwärts kompatibel für Programme, die diese Anweisungen umfassen.

Tabelle 7- 65 Anweisungen FeldLesen und FeldSchreiben

KOP/FUP	SCL	Beschreibung
	<pre>value := member[index];</pre>	FeldLesen liest das Array-Element mit dem Indexwert INDEX aus dem Array, dessen erstes Element vom Parameter MEMBER angegeben wird. Der Wert des Array-Elements wird an die vom Parameter VALUE angegebene Adresse übertragen.
	<pre>member[index] := value;</pre>	FeldSchreiben überträgt den Wert an der vom Parameter VALUE angegebenen Adresse zu dem Array, dessen erstes Element vom Parameter MEMBER angegeben wird. Der Wert wird zu dem Array-Element übertragen, dessen Array-Index vom Parameter INDEX angegeben wird.

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 66 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
Index	Eingang	DInt	Die Indexnummer des Array-Elements, das gelesen oder in das geschrieben werden soll.
Member <sup>1</sup>	Eingang	Array-Elementtypen: Bool, Byte, Word, DWord, Char, SInt, Int, Dint, USInt, UInt, UDInt, Real, LReal	Adresse des ersten Elements in einem eindimensionalen Array, definiert in einem globalen Datenbaustein oder in einer Bausteinschnittstelle.  Beispiel: Wird der Array-Index angegeben als [-2..4], dann ist der Index des ersten Elements -2 und nicht 0.
Value <sup>1</sup>	Ausgang	Bool, Byte, Word, DWord, Char, SInt, Int, Dint, USInt, UInt, UDInt, Real, LReal	Adresse, an die das angegebene Array-Element kopiert wird (FeldLesen). Adresse des Werts, der in das angegebene Array-Element kopiert wird (FeldSchreiben).

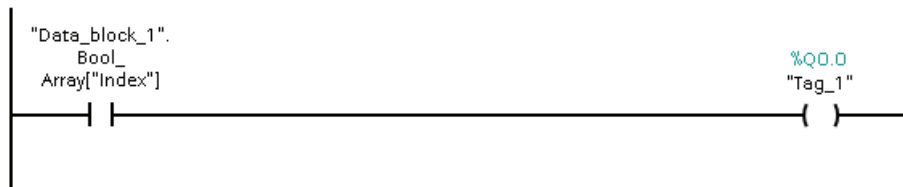
<sup>1</sup> Der Datentyp des vom Parameter MEMBER angegebenen Array-Elements und der Parameter VALUE müssen denselben Datentyp haben.

Der Freigabeausgang ENO ist gleich 0, wenn eine der folgenden Bedingungen zutrifft:

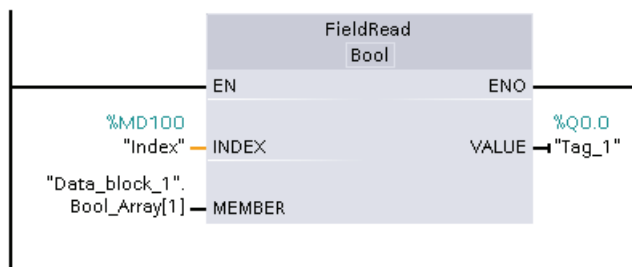
- Der Eingang EN ist im Signalzustand 0.
- Das vom Parameter INDEX angegebene Array-Element ist im vom Parameter MEMBER angegebenen Array nicht definiert.
- Während der Verarbeitung treten Fehler auf, z. B. Überlauf.

### Zugriff auf Daten über Array-Indexierung

Um mit einer Variable auf Elemente eines Arrays zuzugreifen, verwenden Sie einfach die Variable als Array-Index in Ihrer Programmlogik. Im folgenden Netzwerk beispielsweise wird basierend auf dem Booleschen Wert eines Arrays aus Booleschen Werten ein Ausgang in "Datenbaustein\_1", der von der PLC-Variablen "Index" angegeben wird, gesetzt.



Die Logik mit dem Variablen-Array-Index entspricht dem bisherigen Verfahren über die Anweisung FeldLesen:



Die Anweisungen FeldLesen und FeldSchreiben können durch Indexierungslogik mit Variablen-Arrays ersetzt werden.

In SCL gibt es keine Anweisungen FeldLesen oder FeldSchreiben, es wird jedoch die indirekte Adressierung eines Arrays mit einer Variablen unterstützt:

```
#Tag_1 := "Data_block_1".Bool_Array[#Index];
```

### 7.6.3 Operationen Speicher mit Bitmuster belegen

Tabelle 7- 67 Anweisungen FILL\_BLK und UFILL\_BLK


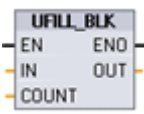
KOP/FUP	SCL	Beschreibung
	<pre>FILL_BLK (   in:=_variant_in,   count:=int,   out=&gt;_variant_out);</pre>	Anweisung Unterbrechbar befüllen: Befüllt einen Adressbereich mit Kopien eines angegebenen Datenelements.
	<pre>UFILL_BLK (   in:=_variant_in,   count:=int   out=&gt;_variant_out);</pre>	Anweisung Ununterbrechbar befüllen: Befüllt einen Adressbereich mit Kopien eines angegebenen Datenelements.

Tabelle 7- 68 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	Datenquelladresse
COUNT	USInt, UInt	Anzahl der zu kopierenden Datenelemente
OUT	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	Datenzieladresse

#### Hinweis

##### Regeln für die Anweisungen Speicher mit Bitmuster belegen

- Um Daten vom Datentyp BOOL zu kopieren, verwenden Sie SET\_BF, RESET\_BF, R, S oder eine Ausgangsspule (KOP).
- Um den Speicher mit einem einzelnen elementaren Datentyp zu belegen, verwenden Sie MOVE.
- Um ein Feld mit einem elementaren Datentyp zu belegen, verwenden Sie FILL\_BLK oder UFILL\_BLK.
- Um ein einzelnes Zeichen in einer Zeichenkette zu kopieren, verwenden Sie MOVE.
- Die Anweisungen FILL\_BLK und UFILL\_BLK können Sie nicht verwenden, um Felder in den Speicherbereichen E, A oder M zu belegen.

Mit den Anweisungen FILL\_BLK und UFILL\_BLK wird das Quelldatenelement IN in die Zieladresse kopiert, wobei die Anfangsadresse im Parameter OUT festgelegt ist. Das Kopieren läuft so lange in den jeweils benachbarten Adressbereichen weiter, bis die Anzahl der Kopien mit dem Wert des Parameters COUNT übereinstimmt.

Die Anweisungen FILL\_BLK und UFILL\_BLK unterscheiden sich in der Verarbeitung von Alarmen:

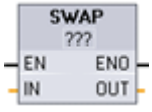
- Alarmereignisse werden **in die Warteschlange gestellt** und während der Ausführung von FILL\_BLK verarbeitet. Die Anweisung FILL\_BLK nutzen Sie, wenn die Daten an der Zieladresse der Übertragung nicht in einem Unterprogramm eines Alarm-OBs verwendet werden, oder, sofern sie verwendet werden, die Zieldaten nicht konsistent sein müssen.
- Alarmereignisse werden **in die Warteschlange gestellt, aber erst verarbeitet**, wenn die Ausführung von UFILL\_BLK beendet ist. Die Anweisung UFILL\_BLK nutzen Sie, wenn die Übertragungsanweisung beendet und die Zieldaten konsistent sein müssen, bevor das Unterprogramm eines Alarm-OBs ausgeführt wird.

Tabelle 7- 69 ENO-Status

ENO	Bedingung	Ergebnis
1	Kein Fehler	Das Element IN wurde erfolgreich in alle COUNT-Ziele kopiert.
0	Der Zielbereich (OUT) überschreitet den verfügbaren Speicherbereich	Elemente, die passen, werden kopiert. Es werden keine Teilelemente kopiert.

### 7.6.4 Operation SWAP (Bytes tauschen)

Tabelle 7- 70 Anweisung SWAP

KOP/FUP	SCL	Beschreibung
	<pre>out := SWAP(in);</pre>	Kehrt die Bytereihenfolge bei Zwei-Byte- und Vier-Byte-Datenelementen um. Die Bitfolge innerhalb eines Bytes wird nicht geändert. ENO ist nach der Ausführung der Anweisung SWAP immer WAHR.

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 71 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Word, DWord	Normal angeordnete Datenbytes in Parameter IN
OUT	Word, DWord	Vertauschte Reihenfolge der Datenbytes in Parameter OUT

Beispiel 1	Parameter IN = MB0 (vor Ausführung)		Parameter OUT = MB4, (nach Ausführung)	
Adresse	MB0	MB1	MB4	MB5
W#16#1234	12	34	34	12
WORD	MSB	LSB	MSB	LSB

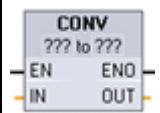


Beispiel 2	Parameter IN = MB0 (vor Ausführung)				Parameter OUT = MB4, (nach Ausführung)			
	MB0	MB1	MB2	MB3	MB4	MB5	MB6	MB7
Adresse	MB0	MB1	MB2	MB3	MB4	MB5	MB6	MB7
DW#16# 12345678	12	34	56	78	78	56	34	12
DWORD	MSB			LSB	MSB			LSB

## 7.7 Umwandeln

### 7.7.1 Anweisung CONV

Tabelle 7- 72 Anweisung Umwandeln (CONV)

KOP/FUP	SCL	Beschreibung
	<pre>out := &lt;data type in&gt;_TO_&lt;data type out&gt;(in);</pre>	Konvertiert ein Datenelement von einem Datentyp in einen anderen Datentyp.

- 1 In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.
- 2 In SCL: Konstruieren Sie die Umwandlungsanweisung durch Angeben des Datentyps für den Eingangsparameter (in) und den Ausgangsparameter (out). Beispiel: DWORD\_TO\_REAL konvertiert einen Doppelwortwert in einen Realzahlenwert.

Tabelle 7- 73 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Bitzeichenkette <sup>1</sup> , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32	Eingangswert
OUT	Bitzeichenkette <sup>1</sup> , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCE32	Eingangswert, umgewandelt in einen neuen Datentyp

- 1 Für die Anweisung dürfen keine Bitzeichenketten (Byte, Word, DWord) ausgewählt werden. Um für einen Parameter der Anweisung einen Operanden vom Datentyp Byte, Word oder DWord einzugeben, wählen Sie eine vorzeichenlose Ganzzahl mit der gleichen Bitlänge aus. Beispiel: Für ein Byte wählen Sie USInt, für ein Word wählen Sie UInt und für DWord wählen Sie UDInt.

Nach Auswahl des Quelldatentyps (Konvertieren aus) wird in der Klappliste eine Reihe möglicher Umwandlungen (Konvertieren in) angezeigt. Umwandlungen von und in BCD16 sind auf den Datentyp Int begrenzt. Umwandlungen von und in BCD32 sind auf den Datentyp DInt begrenzt.

Tabelle 7- 74 ENO-Status

ENO	Beschreibung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN
0	Ergebnis überschreitet den gültigen Bereich für den Datentyp von OUT	OUT wird auf das niederwertigste Byte von IN gesetzt

## 7.7.2 Umwandlungsanweisungen in SCL

### Umwandlungsanweisungen in SCL

Tabelle 7- 75 Umwandlung von Bool, Byte, Word oder DWord

Datentyp	Anweisung	Ergebnis
Bool	BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT	Der Wert wird in das niederwertigste Bit des Zieldatentyps übertragen.
Byte	BYTE_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	BYTE_TO_WORD, BYTE_TO_DWORD	Der Wert wird in das untere Byte des Zieldatentyps übertragen.
	BYTE_TO_SINT, BYTE_TO_USINT	Der Wert wird in den Zieldatentyp übertragen.
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	Der Wert wird in das niederwertigste Byte des Zieldatentyps übertragen.
Word	WORD_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	WORD_TO_BYTE	Das untere Byte des Quellwerts wird in den Zieldatentyp übertragen.
	WORD_TO_DWORD	Der Wert wird in das untere Byte des Zieldatentyps übertragen.
	WORD_TO_SINT, WORD_TO_USINT	Das untere Byte des Quellwerts wird in den Zieldatentyp übertragen.
	WORD_TO_INT, WORD_TO_UINT	Der Wert wird in den Zieldatentyp übertragen.
	WORD_TO_DINT, WORD_TO_UDINT	Der Wert wird in das untere Byte des Zieldatentyps übertragen.
DWord	DWORD_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT, DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	Das untere Byte des Quellwerts wird in den Zieldatentyp übertragen.
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	Der Wert wird in den Zieldatentyp übertragen.

Tabelle 7- 76 Umwandlung einer kurzen Ganzzahl (SInt oder USInt)

Datentyp	Anweisung	Ergebnis
SInt	SINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	SINT_TO_BYTE	Der Wert wird in den Zieldatentyp übertragen.
	SINT_TO_WORD, SINT_TO_DWORD, SINT_TO_INT, SINT_TO_DINT	Der Wert wird in das untere Byte des Zieldatentyps übertragen.
	SINT_TO_USINT, SINT_TO_UINT, SINT_TO_UDINT, SINT_TO_REAL, SINT_TO_LREAL, SINT_TO_CHAR, SINT_TO_STRING	Der Wert wird umgewandelt.
USInt	USINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	USINT_TO_BYTE	Der Wert wird in den Zieldatentyp übertragen.
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	Der Wert wird in das untere Byte des Zieldatentyps übertragen.
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	Der Wert wird umgewandelt.

Tabelle 7- 77 Umwandlung einer Ganzzahl (Int oder UInt)

Datentyp	Anweisung	Ergebnis
Int	INT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	INT_TO_BYTE, INT_TO_DWORD, INT_TO_SINT, INT_TO_USINT, INT_TO_UINT, INT_TO_UDINT, INT_TO_REAL, INT_TO_LREAL, INT_TO_CHAR, INT_TO_STRING	Der Wert wird umgewandelt.
	INT_TO_WORD	Der Wert wird in den Zieldatentyp übertragen.
	INT_TO_DINT	Der Wert wird in das untere Byte des Zieldatentyps übertragen.
UInt	UINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	Der Wert wird umgewandelt.
	UINT_TO_WORD, UINT_TO_DATE	Der Wert wird in den Zieldatentyp übertragen.
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	Der Wert wird in das untere Byte des Zieldatentyps übertragen.

Tabelle 7- 78 Umwandlung einer doppelten Ganzzahl (Dint oder UDInt)

Datentyp	Anweisung	Ergebnis
Dint	DINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	DINT_TO_BYTE, DINT_TO_WORD, DINT_TO_SINT, DINT_TO_USINT, DINT_TO_INT, DINT_TO_UINT, DINT_TO_UDINT, DINT_TO_REAL, DINT_TO_LREAL, DINT_TO_CHAR, DINT_TO_STRING	Der Wert wird umgewandelt.
	DINT_TO_DWORD, DINT_TO_TIME	Der Wert wird in den Zieldatentyp übertragen.
UDInt	UDINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	UDINT_TO_BYTE, UDINT_TO_WORD, UDINT_TO_SINT, UDINT_TO_USINT, UDINT_TO_INT, UDINT_TO_UINT, UDINT_TO_DINT, UDINT_TO_REAL, UDINT_TO_LREAL, UDINT_TO_CHAR, UDINT_TO_STRING	Der Wert wird umgewandelt.
	UDINT_TO_DWORD, UDINT_TO_TOD	Der Wert wird in den Zieldatentyp übertragen.

Tabelle 7- 79 Umwandlung einer Realzahl (Real oder LReal)

Datentyp	Anweisung	Ergebnis
Real	REAL_TO_DWORD, REAL_TO_LREAL	Der Wert wird in den Zieldatentyp übertragen.
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	Der Wert wird umgewandelt.
LReal	LREAL_TO_SINT, LREAL_TO_USINT, LREAL_TO_INT, LREAL_TO_UINT, LREAL_TO_DINT, LREAL_TO_UDINT, LREAL_TO_REAL, LREAL_TO_STRING	Der Wert wird umgewandelt.

Tabelle 7- 80 Umwandlung von Time, DTL, TOD oder Date



Datentyp	Anweisung	Ergebnis
Time	TIME_TO_DINT	Der Wert wird in den Zieldatentyp übertragen.
DTL	DTL_TO_DATE, DTL_TO_TOD	Der Wert wird umgewandelt.
TOD	TOD_TO_UDINT	Der Wert wird umgewandelt.
Date	DATE_TO_UINT	Der Wert wird umgewandelt.

Tabelle 7- 81 Umwandlung von Char oder String

Datentyp	Anweisung	Ergebnis
Char	CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT	Der Wert wird umgewandelt.
	CHAR_TO_STRING	Der Wert wird zum ersten Zeichen der Zeichenkette übertragen.
String	STRING_TO_SINT, STRING_TO_USINT, STRING_TO_INT, STRING_TO_UINT, STRING_TO_DINT, STRING_TO_UDINT, STRING_TO_REAL, STRING_TO_LREAL	Der Wert wird umgewandelt.
	STRING_TO_CHAR	Das erste Zeichen der Zeichenkette wird in Char kopiert.

### 7.7.3 Operationen ROUND (Zahl runden) und TRUNC (Ganzzahl erzeugen)

Tabelle 7- 82 Anweisungen ROUND und TRUNC

KOP/FUP	SCL	Beschreibung
	<pre>out := ROUND (in);</pre>	<p>konvertiert eine Realzahl in eine Ganzzahl. Der Standarddatentyp ist DINT. Wenn ein anderer gültiger Datentyp und nicht DINT ausgegeben wird, muss er explizit deklariert werden, Beispiel: ROUND_REAL oder ROUND_LREAL.</p> <p>Die Nachkommastellen der Realzahl werden auf den nächsten ganzzahligen Wert gerundet (IEEE- aufrunden). Wenn die Zahl genau die Hälfte der Spanne zwischen zwei Ganzzahlen ist (z. B. 10,5), wird die Zahl auf die gerade Ganzzahl gerundet. Beispiel:</p> <ul style="list-style-type: none"> <li>• ROUND (10.5) = 10</li> <li>• ROUND (11.5) = 12</li> </ul>
	<pre>out := TRUNC(in);</pre>	<p>TRUNC konvertiert eine Realzahl in eine Ganzzahl. Die Nachkommastellen der Realzahl werden auf Null verkürzt (IEEE- runden auf Null).</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" (beim Anweisungsnamen) und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 83 Datentypen für die Parameter

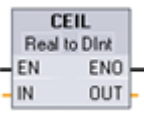

Parameter	Datentyp	Beschreibung
IN	Real, LReal	Eingang Gleitpunktzahl
OUT	Sint, Int, Dint, USInt, UInt, UDIInt, Real, LReal	Gerundeter oder ganzzahliger Ausgang

Tabelle 7- 84 ENO-Status

ENO	Beschreibung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN

#### 7.7.4 Operationen CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen) und FLOOR (Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen)

Tabelle 7- 85 Anweisungen CEIL und FLOOR

KOP/FUP	SCL	Beschreibung
	<code>out := CEIL(in);</code>	Konvertiert eine Realzahl (Real oder LReal) in die nächste Ganzzahl, die größer oder gleich der ausgewählten Realzahl ist (IEEE - Runden auf +unendlich).
	<code>out := FLOOR(in);</code>	Konvertiert eine Realzahl (Real oder LReal) in die nächste Ganzzahl, die kleiner oder gleich der ausgewählten Realzahl ist (IEEE - Runden auf -unendlich).

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" (beim Anweisungsnamen) und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 86 Datentypen für die Parameter

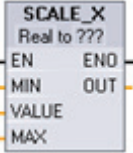
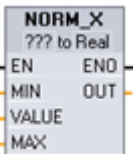
Parameter	Datentyp	Beschreibung
IN	Real, LReal	Eingang Gleitpunktzahl
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Umgewandelter Ausgang

Tabelle 7- 87 ENO-Status

ENO	Beschreibung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN

## 7.7.5 Operationen Skalieren und Normalisieren

Tabelle 7- 88 Anweisungen SCALE\_X und NORM\_X

KOP/FUP	SCL	Beschreibung
	<pre>out :=SCALE_X(min:=_in_,                value:=_in_,                max:=_in_);</pre>	<p>Skaliert den normalisierten Realparameter VALUE (<math>0,0 \leq \text{VALUE} \leq 1,0</math>) in den mit den Parametern MIN und MAX vorgegebenen Datentyp und Wertebereich:  <math>\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}</math></p>
	<pre>out :=NORM_X(min:=_in_,               value:=_in_,               max:=_in_);</pre>	<p>Normalisiert den Parameter VALUE innerhalb des von den Parametern MIN und MAX angegebenen Wertebereichs:  <math>\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN})</math>,  dabei ist (<math>0,0 \leq \text{OUT} \leq 1,0</math>)</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 89 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
MIN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Eingang Mindestwert des Bereichs
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Eingangswert für Skalierung oder Normierung
MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Eingang Höchstwert des Bereichs
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal NORM_X: Real, LReal	Skalierter oder normierter Ausgangswert

<sup>1</sup> Bei SCALE\_X: Die Parameter MIN, MAX und OUT müssen denselben Datentyp haben.  
Bei NORM\_X: Die Parameter MIN, VALUE und MAX müssen denselben Datentyp haben.

**Hinweis**

**Der Parameter VALUE von SCALE\_X muss im Bereich (0,0 <= VALUE <= 1,0) liegen.**

Falls der Parameter VALUE kleiner als 0,0 oder größer als 1,0 ist:

- Die lineare Skalierungsanweisung kann Ausgabewerte OUT erzeugen, die kleiner als der Wert des Parameters MIN oder größer als der Wert des Parameters MAX sind, sofern es sich um OUT-Werte handelt, die im Wertebereich des Datentyps von OUT liegen. Bei der Ausführung von SCALE\_X wird dann ENO = WAHR gesetzt.
- Es ist möglich, skalierte Zahlen zu erzeugen, die nicht im Bereich des Datentyps von OUT liegen. Dann wird der Wert des Parameters OUT auf einen Zwischenwert gesetzt, der dem niederwertigsten Anteil der skalierten Realzahl vor der endgültigen Umwandlung in den Datentyp von OUT entspricht. Bei der Ausführung von SCALE\_X wird dann ENO = FALSCH gesetzt.

**Der Parameter VALUE von NORM\_X muss im Bereich (MIN <= VALUE <= MAX) liegen.**

Wenn der Parameter VALUE kleiner als MIN oder größer als MAX ist, kann die lineare Skalierungsanweisung normalisierte Ausgabewerte OUT erzeugen, die kleiner als 0,0 oder größer als 1,0 sind. Bei der Ausführung von NORM\_X wird dann ENO = WAHR gesetzt.

Tabelle 7- 90 ENO-Status

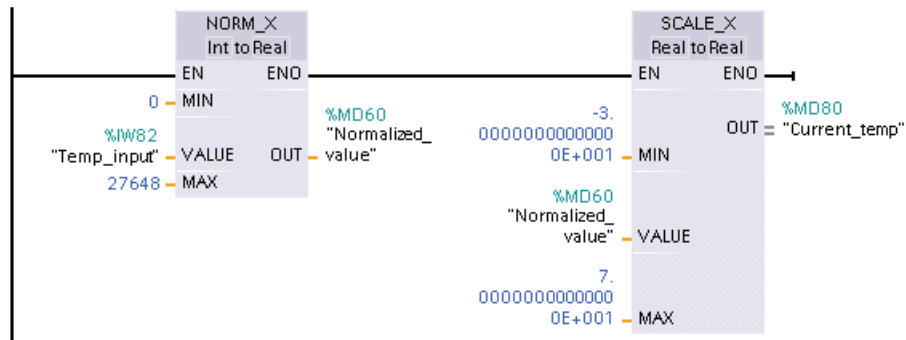
ENO	Bedingung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	Das Ergebnis überschreitet den gültigen Bereich für den Datentyp von OUT.	Zwischenergebnis: Der niederwertigste Anteil einer Realzahl vor der endgültigen Umwandlung in den Datentyp von OUT.
0	ParameterMAX <= MIN	SCALE_X: Der niederwertigste Anteil der Realzahl VALUE, mit dem die Größe von OUT belegt wird. NORM_X: VALUE im Datentyp VALUE, zur Belegung der Doppelwortgröße erweitert.
0	Parameter VALUE = +/- INF oder +/- NaN	VALUE wird in OUT geschrieben

**Beispiel (KOP): Normalisieren und Skalieren eines Analogeingangswerts**

Ein Analogeingang eines analogen Signalmoduls oder Signalboards mit Stromeingang liegt im Bereich von 0 bis 27648 der gültigen Werte. Angenommen, ein Analogeingang stellt eine Temperatur dar, wobei der Wert 0 des Analogeingangs -30,0 Grad C und der Wert 27648 die Temperatur 70,0 Grad C darstellt.



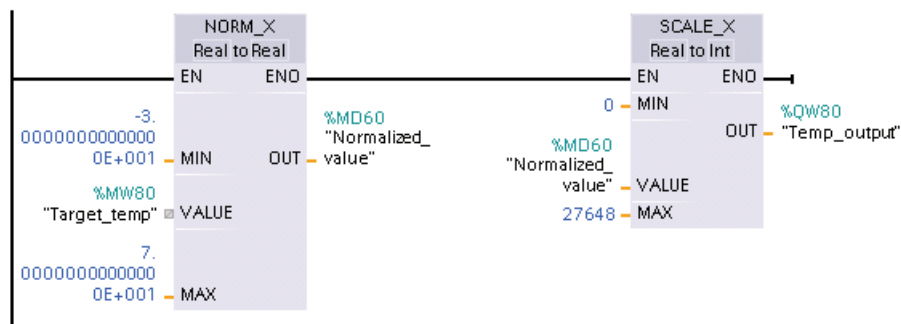
Um den Analogwert in die entsprechenden physikalischen Einheiten umzuwandeln, normalisieren Sie den Eingang in einen Wert zwischen 0,0 und 1,0 und skalieren ihn dann zwischen -30,0 und 70,0. Der resultierende Wert ist die vom Analogeingang dargestellte Temperatur in Grad C:



Beachten Sie, dass, wenn der Analogeingang von einem analogen Signalmodul oder Signalboard mit Spannungseingang kommen würde, der MIN-Wert für die Anweisung NORM\_X -27648 und nicht 0 wäre.

### Beispiel (KOP): Normalisieren und Skalieren eines Analogausgangswerts

Ein in einem analogen Signalmodul oder Signalboard mit Stromausgang zu setzender Analogausgang muss im Bereich von 0 bis 27648 der gültigen Werte liegen. Angenommen, ein Analogausgang stellt eine Temperatureinstellung dar, wobei der Wert 0 des Analogeingangs -30,0 Grad C und der Wert 27648 die Temperatur 70,0 Grad C darstellt. Um im Speicher einen Temperaturwert, der zwischen -30,0 und 70,0 liegt, in einen Wert für den Analogausgang im Bereich zwischen 0 und 27648 umzuwandeln, müssen Sie den Wert in physikalischen Einheiten in einen Wert zwischen 0,0 und 1,0 normalisieren und ihn dann in den Bereich des Analogausgangs zwischen 0 und 27648 skalieren:



Beachten Sie, dass, wenn der Analogausgang für ein analoges Signalmodul oder Signalboard mit Spannungseingang vorgesehen wäre, der MIN-Wert für die Anweisung SCALE\_X -27648 und nicht 0 wäre.

Weitere Informationen zur Darstellung von Analogeingängen (Seite 818) und zur Darstellung von Analogausgängen (Seite 819) für Spannung und Strom finden Sie in den technischen Daten.

## 7.8 Programmsteuerung

### 7.8.1 Übersicht über die Programmsteuerungsanweisungen in SCL

Structured Control Language (SCL) bietet drei Arten von Programmsteuerungsanweisungen für die Strukturierung Ihres Anwenderprogramms:

- **Auswahl**anweisungen: Mit Hilfe einer Auswahlanweisung können Sie den Programmfluss in alternative Anweisungsfolgen steuern.
- **Schleifen**: Sie können die Schleifenausführung über Wiederholungsanweisungen steuern. Eine Wiederholungsanweisung gibt an, welche Teile eines Programm abhängig von bestimmten Bedingungen wiederholt werden sollen.
- **Programmsprünge**: Ein Programmsprung bewirkt einen direkten Sprung zu einem angegebenen Sprungziel und damit zu einer anderen Anweisung in demselben Baustein.

Die Programmsteuerungsanweisungen verwenden die Syntax der Programmiersprache PASCAL.

Tabelle 7- 91 Arten von Programmsteuerungsanweisungen in SCL

Programmsteuerungsanweisung		Beschreibung
Auswahl	IF-THEN-Anweisung (Seite 235)	Ermöglicht Ihnen, den Programmfluss in Abhängigkeit von einer Bedingung, die entweder WAHR oder FALSCH ist, in eine von zwei Alternativen zu verzweigen.
	CASE-Anweisung (Seite 236)	Ermöglicht die selektive Ausführung einer von $n$ alternativen Verzweigungen, basierend auf dem Wert einer Variablen.
Schleife	FOR-Anweisung (Seite 237)	Dient zur Wiederholung einer Folge von Anweisungen, solange die Steuervariable innerhalb des angegebenen Wertebereichs liegt.
	WHILE-DO-Anweisung (Seite 238)	Dient zur Wiederholung einer Folge von Anweisungen, solange eine Durchführungsbedingung erfüllt ist.
	REPEAT-UNTIL-Anweisung (Seite 239)	Dient zur Wiederholung einer Folge von Anweisungen, bis eine Abbruchbedingung erfüllt ist.
Programmsprung	CONTINUE-Anweisung (Seite 240)	Dient zum Abbruch der Ausführung des momentanen Schleifendurchlaufs.
	EXIT-Anweisung (Seite 241)	Dient zum Verlassen einer Schleife an beliebiger Stelle und unabhängig vom Erfülltsein der Abbruchbedingung.
	GOTO-Anweisung (Seite 241)	Bewirkt den sofortigen Sprung zu einer angegebenen Sprungmarke.
	IF-THEN-Anweisung (Seite 235)	Bewirkt das Verlassen des gerade ausgeführten Codebausteins und die Rückkehr zum aufrufenden Codebaustein.

#### Siehe auch

RETURN-Anweisung (Seite 242)

## 7.8.2 IF-THEN-Anweisung

Die IF-THEN-Anweisung ist eine bedingte Anweisung, die den Programmfluss steuert, indem eine Gruppe von Anweisungen basierend auf der Auswertung eines Bool-Werts eines logischen Ausdrucks ausgeführt wird. Die Ausführung mehrerer IF-THEN-Anweisungen können Sie auch mit Hilfe von Klammern verschachteln bzw. strukturieren.

Tabelle 7- 92 Elemente der IF-THEN-Anweisung

SCL	Beschreibung
<pre>IF "Bedingung" THEN   Anweisung_A;   Anweisung_B;   Anweisung_C; ;</pre>	<p>Falls "Bedingung" WAHR oder 1 ist, dann die folgenden Anweisungen bis zur Anweisung END_IF ausführen.</p> <p>Falls "Bedingung" FALSCH oder 0 ist, dann bis zur Anweisung END_IF überspringen (es sei denn, das Programm enthält optionale ELSIF- oder ELSE-Anweisungen).</p>
<pre>[ELSIF "Bedingung-n" THEN   Anweisung_N; ;]</pre>	<p>Die optionale ELSEIF<sup>1</sup>-Anweisung bietet weitere auszuwertende Bedingungen. Beispiel: Falls "Bedingung" in der IF-THEN-Anweisung FALSCH ist, wertet das Programm "Bedingung-n" aus. Falls "Bedingung-n" WAHR ist, dann "Anweisung_N" ausführen.</p>
<pre>[ELSE   Anweisung_X; ;]</pre>	<p>Die optionale ELSE-Anweisung bietet Anweisungen, die auszuführen sind, wenn die "Bedingung" der IF-THEN-Anweisung FALSCH ist.</p>
<pre>END_IF;</pre>	<p>Die END_IF-Anweisung beendet die IF-THEN-Anweisung.</p>

<sup>1</sup> Sie können in eine IF-THEN-Anweisung mehrere ELSIF-Anweisungen einfügen.

Tabelle 7- 93 Variablen der IF-THEN-Anweisung

Variablen	Beschreibung
"Bedingung"	Erforderlich. Der logische Ausdruck ist entweder WAHR (1) oder FALSCH (0).
"Anweisung_A"	Optional. Eine oder mehrere Anweisungen, die auszuführen sind, wenn "Bedingung" WAHR ist.
"Bedingung-n"	Optional. Der logische Ausdruck, der von der optionalen ELSIF-Anweisung auszuwerten ist.
"Anweisung_N"	Optional. Eine oder mehrere Anweisungen, die auszuführen sind, wenn "Bedingung-n" der ELSIF-Anweisung WAHR ist.
"Anweisung_X"	Optional. Eine oder mehrere Anweisungen, die auszuführen sind, wenn "Bedingung" der IF-THEN-Anweisung FALSCH ist.

Eine IF-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Die erste Anweisungsfolge, deren logischer Ausdruck = WAHR ist, wird ausgeführt. Die übrigen Anweisungsfolgen werden nicht ausgeführt.
- Wenn kein Boolescher Ausdruck = WAHR ist, wird die von ELSE eingeführte Anweisungsfolge ausgeführt (oder keine Anweisungsfolge, falls keine ELSE-Verzweigung vorhanden ist).
- Die Anzahl der ELSIF-Anweisungen ist nicht begrenzt.

---

#### Hinweis

Die Verwendung einer oder mehrerer ELSIF-Verzweigungen hat gegenüber einer Folge von IF-Anweisungen den Vorteil, dass die logischen Ausdrücke, die auf einen gültigen Ausdruck folgen, nicht mehr ausgewertet werden. Die Laufzeit eines Programms kann dadurch verringert werden.

---

### 7.8.3 CASE-Anweisung

Tabelle 7- 94 Elemente der CASE-Anweisung

SCL	Beschreibung
<pre> CASE "Testwert" OF     "Werteliste": Anweisung[; Anweisung, ...]     "Werteliste": Anweisung[; Anweisung, ...] [ELSE Else-Anweisung[; Else-Anweisung, ...]] END CASE;</pre>	Die Anweisung CASE führt eine von mehreren Anweisungsgruppen abhängig vom Wert eines Ausdrucks aus.

Tabelle 7- 95 Parameter

Parameter	Beschreibung
"Testwert"	Erforderlich. Beliebiger numerischer Ausdruck vom Datentyp Int.
"Werteliste"	Erforderlich. Ein einzelner Wert oder eine Liste mit durch Komma getrennten Werten oder Wertebereichen. (Einen Wertebereich geben Sie mittels zwei Punkten an: 2..8). Das folgende Beispiel zeigt die verschiedenen Varianten von Wertelisten: 1: Anweisung_A; 2, 4: Anweisung_B; 3, 5..7,9: Anweisung_C;
Anweisung	Erforderlich. Eine oder mehrere Anweisungen, die ausgeführt werden, wenn "Testwert" einem Wert in der Werteliste entspricht.
Else-Anweisung	Optional. Eine oder mehrere Anweisungen, die ausgeführt werden, wenn kein Wert der "Werteliste" eine Übereinstimmung ergibt.

Eine CASE-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Der Auswahlausdruck muss einen Wert vom Typ Int zurückgeben.
- Bei der Verarbeitung einer CASE-Anweisung prüft das Programm, ob der Wert des Auswahlausdrucks in einer angegebenen Werteliste enthalten ist. Wird eine Übereinstimmung gefunden, wird die der Liste zugewiesene Anweisungskomponente ausgeführt.
- Wird keine Übereinstimmung gefunden, wird der auf ELSE folgende Programmabschnitt ausgeführt. Falls keine ELSE-Verzweigung vorhanden ist, wird keine Anweisung ausgeführt.

CASE-Anweisungen können verschachtelt werden. Jede verschachtelte CASE-Anweisung benötigt eine zugehörige END\_CASE-Anweisung.

```

CASE var1 OF
    1 : var2 := "A";
    2 : var2 := "B";
ELSE
    CASE var3 OF
        65..90: var2 := "Großbuchstabe";
        97..122: var2 := "Kleinbuchstabe";
    ELSE
        var2:= "Sonderzeichen";
    END_CASE;
END_CASE;

```

## 7.8.4 FOR-Anweisung

Tabelle 7- 96 Elemente der FOR-Anweisung

SCL	Beschreibung
<pre> FOR "Steuervariable" := "Anfang" TO "Ende" [BY "Inkrement"] DO     Anweisung; ; END_FOR; </pre>	<p>Eine FOR-Anweisung dient zur Wiederholung einer Folge von Anweisungen, solange eine Steuervariable innerhalb des angegebenen Wertebereichs liegt. Die Definition einer Schleife mit FOR umfasst die Angabe eines Anfangs- und eines Endwerts. Beide Werte müssen denselben Typ haben wie die Steuervariable.</p>

Tabelle 7- 97 Parameter

Parameter	Beschreibung
"Steuervariable"	Erforderlich. Eine Ganzzahl (Int oder DInt), die als Schleifenzähler dient.
"Anfang"	Erforderlich. Einfacher Ausdruck, der den Anfangswert der Steuervariablen angibt.

Parameter	Beschreibung
"Ende"	Erforderlich. Einfacher Ausdruck, der den Abschlusswert der Steuervariablen angibt.
"Inkrement"	Optional. Betrag, um den eine "Steuervariable" nach jeder Schleife geändert wird. Das "Inkrement" hat denselben Datentyp wie "Steuervariable". Wenn der Wert von "Inkrement" nicht angegeben ist, wird der Wert der Variablen nach jeder Schleife um 1 erhöht. Sie können "Inkrement" nicht während der Ausführung der FOR-Anweisung ändern.

Die FOR-Anweisung wird wie folgt ausgeführt:

- Beim Start der Schleife wird die Steuervariable auf den Anfangswert gesetzt (Anfangszuweisung) und bei jedem Schleifendurchlauf um das angegebene Inkrement erhöht (positives Inkrement) oder verringert (negatives Inkrement), bis der Endwert erreicht ist.
- Nach jedem Schleifendurchlauf wird geprüft, ob die Bedingung (Endwert erreicht) erfüllt ist oder nicht. Ist die Bedingung erfüllt, wird die Anweisungsfolge ausgeführt. Andernfalls wird die Schleife und damit die Anweisungsfolge übersprungen.

Regeln für die Formulierung von FOR-Anweisungen:

- Die Steuervariable darf nur vom Datentyp Int oder DInt sein.
- Sie können die Anweisung BY [Inkrement] weglassen. Wenn kein Inkrement angegeben wird, wird automatisch ein Inkrement von +1 angenommen.

Um die Schleife unabhängig vom Zustand des Ausdrucks "Bedingung" zu beenden, verwenden Sie die EXIT-Anweisung (Seite 241). Die EXIT-Anweisung führt die Anweisung aus, die unmittelbar auf die END\_FOR-Anweisung folgt.

Mit der CONTINUE-Anweisung (Seite 240) überspringen Sie die nachfolgenden Anweisungen einer FOR-Schleife und setzen die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist.

## 7.8.5 WHILE-DO-Anweisung

Tabelle 7- 98 WHILE-Anweisung

SCL	Beschreibung
<pre>WHILE "Bedingung" DO   Anweisung;   Anweisung;   ...; END WHILE;</pre>	<p>Die WHILE-Anweisung führt eine Anweisungsfolge aus, bis eine vorgegebene Bedingung WAHR ist.</p> <p>WHILE-Schleifen können Sie verschachteln. Die END_WHILE-Anweisung bezieht sich auf die zuletzt ausgeführte WHILE-Anweisung.</p>

Tabelle 7- 99 Parameter

Parameter	Beschreibung
"Bedingung"	Erforderlich. Ein logischer Ausdruck, dessen Auswertung WAHR oder FALSCH ergibt. (Die Bedingung "Null" wird als FALSCH ausgewertet.)
Anweisung	Optional. Eine oder mehrere Anweisungen, die ausgeführt werden, bis die Bedingung WAHR ist.

**Hinweis**

Die WHILE-Anweisung wertet den Zustand der "Bedingung" aus, bevor die Anweisungen ausgeführt werden. Um die Anweisungen unabhängig vom Zustand der "Bedingung" mindestens einmal auszuführen, verwenden Sie die REPEAT-Anweisung.

Die WHILE-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Vor jeder Ausführung des Schleifeninhalts, wird die Ausführungsbedingung ausgewertet.
- Der auf DO folgende Schleifeninhalt wird solange wiederholt, bis die Ausführungsbedingung den Wert WAHR annimmt.
- Tritt der Wert FALSCH auf, wird die Schleife übersprungen und die auf die Schleife folgende Anweisung wird ausgeführt.

Um die Schleife unabhängig vom Zustand des Ausdrucks "Bedingung" zu beenden, verwenden Sie die EXIT-Anweisung (Seite 241). Die EXIT-Anweisung führt die Anweisung aus, die unmittelbar auf die END\_WHILE-Anweisung folgt.

Mit der CONTINUE-Anweisung überspringen Sie die nachfolgenden Anweisungen einer WHILE-Schleife und setzen die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist.

## 7.8.6 REPEAT-UNTIL-Anweisung

Tabelle 7- 100 REPEAT-Anweisung

SCL	Beschreibung
<pre> REPEAT   Anweisung; ; UNTIL "Bedingung" END_REPEAT;</pre>	<p>Die REPEAT-Anweisung führt eine Gruppe von Anweisungen aus, bis eine vorgegebene Bedingung WAHR ist.</p> <p>REPEAT-Schleifen können Sie verschachteln. Die END_REPEAT-Anweisung bezieht sich immer auf die zuletzt ausgeführte REPEAT-Anweisung.</p>

Tabelle 7- 101 Parameter

Parameter	Beschreibung
Anweisung	Optional. Eine oder mehrere Anweisungen, die ausgeführt werden, bis die Bedingung WAHR ist.
"Bedingung"	Erforderlich. Einer oder mehrere Ausdrücke der zwei folgenden Arten: Ein numerischer Ausdruck oder ein Zeichenkettenausdruck, dessen Auswertung WAHR oder FALSCH ergibt. Die Bedingung "Null" wird als FALSCH ausgewertet.

**Hinweis**

Vor der Auswertung des Zustands der "Bedingung" führt die REPEAT-Anweisung die Anweisungen während der ersten Durchführung der Schleife aus (auch wenn die "Bedingung" FALSCH ist). Um den Zustand der "Bedingung" vor der Ausführung der Anweisungen zu prüfen, verwenden Sie die WHILE-Anweisung.

Um die Schleife unabhängig vom Zustand des Ausdrucks "Bedingung" zu beenden, verwenden Sie die EXIT-Anweisung (Seite 241). Die EXIT-Anweisung führt die Anweisung aus, die unmittelbar auf die END\_REPEAT-Anweisung folgt.

Mit der CONTINUE-Anweisung (Seite 240) überspringen Sie die nachfolgenden Anweisungen einer REPEAT-Schleife und setzen die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist.

### 7.8.7 CONTINUE-Anweisung

Tabelle 7- 102 CONTINUE-Anweisung

SCL	Beschreibung
<b>CONTINUE</b> Anweisung; ;	Die CONTINUE-Anweisung überspringt die nachfolgenden Anweisungen einer Programmschleife (FOR, WHILE, REPEAT) und setzt die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist. Ist dies nicht der Fall, wird die Schleife fortgesetzt.

Die CONTINUE-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Diese Anweisung beendet die Ausführung des Schleifeninhalts sofort.
- Abhängig davon, ob die Bedingung für die Wiederholung der Schleife erfüllt ist oder nicht, wird der Schleifeninhalt erneut ausgeführt oder die Wiederholungsanweisung beendet und die unmittelbar nachfolgende Anweisung ausgeführt.
- In einer FOR-Anweisung wird die Steuervariable direkt nach einer CONTINUE-Anweisung um das angegebene Inkrement erhöht.

Verwenden Sie die CONTINUE-Anweisung nur innerhalb einer Schleife. In verschachtelten Schleifen bezieht sich die CONTINUE-Anweisung immer auf die Schleife, in der sich die Anweisung befindet. Die CONTINUE-Anweisung wird üblicherweise in Verbindung mit einer IF-Anweisung verwendet.

Wenn die Schleife unabhängig von der Beendigungsprüfung beendet werden soll, verwenden Sie die EXIT-Anweisung.

Das folgende Beispiel zeigt die Verwendung der CONTINUE-Anweisung, mit der beim Berechnen eines Prozentwerts der Fehler "Division durch 0" verhindert werden soll:

```

FOR x = 0 TO 10 DO
IF value[i] = 0 THEN CONTINUE; END_IF;
    p := part / value[i] * 100;
    s := INT_TO_STRING(p);
    percent=CONCAT(IN1:=s, IN2:="%");
END_FOR;
    
```



## 7.8.8 EXIT-Anweisung

Tabelle 7- 103 Anweisung EXIT

SCL	Beschreibung
<b>EXIT;</b>	Die EXIT-Anweisung dient zum Beenden einer Schleife (FOR, WHILE oder REPEAT) an beliebiger Stelle und unabhängig davon, ob die Beendigungsbedingung erfüllt ist.

Die EXIT-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Diese Anweisung bewirkt, dass die Wiederholungsanweisung, die die EXIT-Anweisung unmittelbar umgibt, sofort beendet wird.
- Die Ausführung des Programm wird nach dem Ende der Schleife fortgesetzt (z. B. nach END\_FOR).

Verwenden Sie die EXIT-Anweisung innerhalb einer Schleife. In verschachtelten Schleifen gibt die EXIT-Anweisung die Verarbeitung an die nächsthöhere Schachtelungebene zurück.

```
FOR i = 0 TO 10 DO
CASE Wert[i, 0] OF
  1..10: Wert [i, 1]:="A";
  11..40: Wert [i, 1]:="B";
  41..100: Wert [i, 1]:="C";
ELSE
EXIT;
END_CASE;
END_FOR;
```

## 7.8.9 GOTO-Anweisung

Tabelle 7- 104 GOTO-Anweisung

SCL	Beschreibung
<b>GOTO JumpLabel;</b> <b>Anweisung;</b> <b>... ;</b> <b>JumpLabel: Anweisung;</b>	Die GOTO-Anweisung überspringt Anweisungen, indem sie zu einer Sprungmarke in demselben Baustein springt.  Die Sprungmarke ("JumpLabel") und die GOTO-Anweisung müssen sich in demselben Baustein befinden. Der Name einer Sprungmarke darf innerhalb eines Bausteins nur einmal vergeben werden. Jede Sprungmarke kann Ziel verschiedener GOTO-Anweisungen sein.

Es ist nicht möglich, in einen Schleifenabschnitt zu springen (FOR, WHILE oder REPEAT). Es ist möglich, aus einer Schleife herauszuspringen.

Im folgenden Beispiel gilt: Je nach Wert des Operanden "Variablenwert" wird die Ausführung des Programms an dem Punkt wieder aufgenommen, der von der entsprechenden Sprungmarke vorgegeben wird. Ist "Variablenwert" gleich 2, nimmt das Programm die Ausführung an Sprungmarke "MyLabel2" wieder auf und überspringt "MyLabel1".

```
CASE "Variablenwert" OF
  1 : GOTO MeineBeschriftung1;
  2 : GOTO MeineBeschriftung2;
ELSE GOTO MeineBeschriftung3;
END_CASE;
```

```

MeineBeschriftung1: "Variable_1" := 1;
MeineBeschriftung2: "Variable_2" := 1;
MeineBeschriftung3: "Variable_4" := 1;
    
```

### 7.8.10 RETURN-Anweisung

Tabelle 7- 105 Anweisung RETURN

SCL	Beschreibung
RETURN ;	Die RETURN-Anweisung beendet den Codebaustein, der gerade ausgeführt wird, ohne Bedingungen. Die Programmausführung kehrt zum aufrufenden Baustein oder zum Betriebssystem zurück (beim Beenden eines OB).

Beispiel für eine RETURN-Anweisung:

```

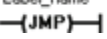

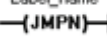

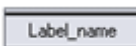

IF "Fehler" <> 0 THEN
RETURN ;
END_IF ;
    
```

#### Hinweis

Nach der Ausführung der letzten Anweisung kehrt der Codebaustein automatisch zum aufrufenden Baustein zurück. Fügen Sie eine RETURN-Anweisung nicht am Ende des Codebausteins ein.

### 7.8.11 Operationen Sprung und Sprungmarke

Tabelle 7- 106 Anweisungen JMP, JMPN und LABEL

KOP	FUP	SCL	Beschreibung
Label_name 	Label_name 	Siehe GOTO (Seite 241)-Anweisung.	Bei Signalfuss zu einer JMP-Spule (KOP) oder wenn der JMP-Boxeingang wahr ist (FUP), wird die Programmausführung mit der ersten Anweisung nach der angegebenen Sprungmarke fortgesetzt.
Label_name 	Label_name 		Ist kein Signalfuss zu einer JMPN-Spule (KOP) vorhanden oder ist der JMPN-Boxeingang falsch (FUP), wird die Programmausführung mit der ersten Anweisung nach der angegebenen Sprungmarke fortgesetzt.
			Sprungziel für eine Sprunganweisung JMP oder JMPN

<sup>1</sup> Sie erstellen die Namen für die Sprungmarken, indem Sie die Anweisung LABEL direkt eingeben. Über das Parametersymbol können Sie die verfügbaren Sprungmarken für die Anweisung JMP oder JMPN auswählen. Sie können den Namen der Sprungmarke auch direkt in die Anweisung JMP oder JMPN eingeben.

Tabelle 7- 107 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Label_name	Kennung der Sprungmarke	Kennzeichnung für Sprunganweisungen und die entsprechende Sprungmarke für das Sprungziel

- Jede Sprungmarke muss innerhalb eines Codebausteins eindeutig sein.
- Sie können innerhalb eines Codebausteins springen, aber Sie können nicht von einem Codebaustein in einen anderen Codebaustein springen.
- Sie können vorwärts oder rückwärts springen.
- Sie können von mehreren Stellen eines Codebausteins zu derselben Sprungmarke springen.

## 7.8.12 Anweisung JMP\_LIST

Tabelle 7- 108 Anweisung JMP\_LIST


KOP/FUP	SCL	Beschreibung
	<pre> CASE k OF   0: GOTO dest0;   1: GOTO dest1;   2: GOTO dest2;   [n: GOTO destn;] END_CASE; </pre>	<p>Die Anweisung JMP_LIST verteilt die Programmsprünge, um die Ausführung von Programmabschnitten zu steuern. Je nach Wert des Eingangs K erfolgt ein Sprung zur entsprechenden Sprungmarke im Programm. Die Programmausführung wird mit den Programmanweisungen fortgesetzt, die auf das Sprungziel folgen. Wenn der Wert des Eingangs K die Anzahl der Sprungmarken - 1 überschreitet, wird kein Sprung durchgeführt und die Verarbeitung wird mit dem nächsten Netzwerk im Programm fortgesetzt.</p>

Tabelle 7- 109 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
K	UInt	Steuerwert für die Sprungverteilung
DEST0, DEST1, ..., DESTn.	Sprungmarken im Programm	<p>Sprungmarken der Sprungziele entsprechend spezifischen Werten des Parameters K:</p> <p>Wenn der Wert von K gleich 0 ist, wird ein Sprung zu der Sprungmarke durchgeführt, die dem Ausgang DEST0 zugewiesen ist. Wenn der Wert von K gleich 1 ist, wird ein Sprung zu der Sprungmarke durchgeführt, die dem Ausgang DEST1 zugewiesen ist usw. Wenn der Wert des Eingangs K (die Anzahl der Sprungmarken - 1) überschreitet, wird kein Sprung durchgeführt und die Verarbeitung wird mit dem nächsten Netzwerk im Programm fortgesetzt.</p>

In KOP und FUP: Beim Einfügen der Box JMP\_LIST in Ihr Programm sind zwei Ausgänge für Sprungmarken vorhanden. Sie können Sprungziele hinzufügen oder löschen.



Klicken Sie in der Box auf das Symbol zum Erstellen (links vom letzten Parameter DEST), um neue Ausgänge für Sprungmarken hinzuzufügen.



- Klicken Sie mit der rechten Maustaste auf einen Ausgangsanschluss und wählen Sie "Ausgang einfügen".
- Klicken Sie mit der rechten Maustaste auf einen Ausgangsanschluss und wählen Sie "Löschen".

### 7.8.13 Anweisung SWITCH

Tabelle 7- 110 Anweisung SWITCH

KOP/FUP	SCL	Beschreibung
	Nicht verfügbar	Die Anweisung SWITCH verteilt die Programmsprünge, um die Ausführung von Programmabschnitten zu steuern. Je nach Ergebnis des Vergleichs zwischen dem Wert des Eingangs K und den Werten der angegebenen Vergleichseingänge wird ein Sprung zu der Sprungmarke durchgeführt, die dem ersten Vergleichstest mit dem Ergebnis WAHR entspricht. Ist keiner der Vergleiche wahr, erfolgt ein Sprung zu der Sprungmarke, die ELSE zugewiesen ist. Die Programmausführung wird mit den Programmanweisungen fortgesetzt, die auf das Sprungziel folgen.

- 1 In KOP und FUP: Klicken Sie unterhalb des Box-Namens und wählen Sie in der Klappliste einen Datentyp aus.
- 2 In SCL: Verwenden Sie Vergleiche über IF-THEN.

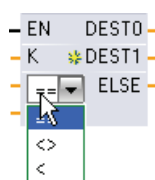
Tabelle 7- 111 Datentypen für die Parameter

Parameter	Datentyp <sup>1</sup>	Beschreibung
K	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Eingang für den allgemeinen Vergleichswert
==, <>, <, <=, >, >=	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Eingänge für einzelne Vergleichswerte für spezifische Vergleichsarten
DEST0, DEST1, ..., DESTn. ELSE	Sprungmarken im Programm	Sprungmarken der Sprungziele entsprechend spezifischen Vergleichen:  Der Vergleichseingang unter und neben dem Eingang K wird zuerst verarbeitet und bewirkt einen Sprung zu der Sprungmarke, die DEST0 zugewiesen ist, wenn der Vergleich zwischen dem Wert K und diesem Eingang wahr ist. Der nächste Vergleichstest verwendet den nächsten Eingang darunter und bewirkt einen Sprung zu der Sprungmarke, die DEST1 zugewiesen ist, wenn der Vergleich wahr ist. Die übrigen Vergleiche werden auf ähnliche Weise verarbeitet und wenn keiner der Vergleiche wahr ist, erfolgt ein Sprung zu der Sprungmarke, die dem Ausgang ELSE zugewiesen ist.

<sup>1</sup> Der Eingang K und die Vergleichseingänge (==, <>, <, <=, >, >=) müssen denselben Datentyp haben.

### Eingänge hinzufügen, Eingänge löschen und Vergleichsarten angeben

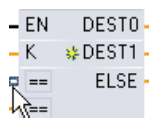
Beim Einfügen der Box SWITCH in KOP oder FUP in Ihrem Programm sind zwei Vergleichseingänge vorhanden. Sie können Vergleichsarten zuweisen und Eingänge bzw. Sprungziele hinzufügen oder löschen (siehe unten).



Klicken Sie in der Box auf einen Vergleichsoperator und wählen Sie in der Klappliste einen neuen Operator aus.



Klicken Sie in der Box auf das Symbol zum Erstellen (links vom letzten Parameter DEST), um neue Parameter für Vergleichsziele hinzuzufügen.



- Klicken Sie mit der rechten Maustaste auf einen Eingangsanschluss und wählen Sie "Eingang einfügen".
- Klicken Sie mit der rechten Maustaste auf einen Eingangsanschluss und wählen Sie "Löschen".

Tabelle 7- 112 Auswahl des Datentyps für die Box SWITCH und zulässige Vergleichsanweisungen

Datentyp	Vergleich	Syntax des Operators
Byte, Word, DWord	Gleich	==
	Ungleich	<>
SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, TOD, Date	Gleich	==
	Ungleich	<>
	Größer oder gleich	>=
	Kleiner oder gleich	<=
	Größer als	>
	Kleiner als	<

**Regeln für die Platzierung der Box SWITCH**

- Vor dem Vergleichseingang darf keine KOP/FUP-Anweisung angeschlossen werden.
- Es gibt keinen ENO-Ausgang, deshalb ist nur eine SWITCH-Anweisung in einem Netzwerk zulässig, und die SWITCH-Anweisung muss die letzte Anweisung im Netzwerk sein.

**7.8.14 Programmsteuerungsoperation RET**

Mit der optionalen Anweisung RET wird die Ausführung des aktuellen Bausteins beendet. Nur bei Signalfluss zur Spule RET (KOP) oder wenn die Box RET wahr ist (FUP), wird die Programmausführung des aktuellen Bausteins an diesem Punkt beendet; nach der Anweisung RET werden keine weiteren Anweisungen ausgeführt. Wenn es sich bei dem aktuellen Baustein um einen OB handelt, wird der Parameter "Return\_Value" ignoriert. Wenn es sich bei dem aktuellen Baustein um eine FC oder einen FB handelt, wird der Wert des Parameters "Return\_Value " als ENO-Wert der aufgerufenen Box wieder an die aufrufende Routine übergeben.

Sie müssen die Anweisung RET nicht als letzte Anweisung in einen Baustein einfügen, dies geschieht automatisch. Sie können mehrere Anweisungen RET in einen einzigen Baustein einfügen.

Für SCL siehe RETURN (Seite 242)-Anweisung.

Tabelle 7- 113 Programmsteuerungsanweisung Rückgabewert (RET)



KOP	FUP	SCL	Beschreibung
		<p>RETURN ;</p>	<p>Beendet die Ausführung des aktuellen Bausteins.</p>

Tabelle 7- 114 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Return_Value	Bool	Der Parameter "Return_value" der Anweisung RET ist dem Ausgang ENO der Bausteinaufruf-Box im aufrufenden Baustein zugewiesen.

Beispiel für die Verwendung einer Anweisung RET in einem Codebaustein FC:


1. Erstellen Sie ein neues Projekt und fügen Sie eine FC ein:
2. Bearbeiten Sie die FC:
  - Fügen Sie Anweisungen aus dem Anweisungsverzeichnis ein.
  - Fügen Sie eine Anweisung RET einschließlich einem der folgenden Elemente für den Parameter "Return\_Value" ein:  
WAHR, FALSCH oder eine Adresse, die den erforderlichen Rückgabewert angibt.
  - Fügen Sie weitere Anweisungen ein.
3. Rufen Sie die FC aus dem Hauptprogramm MAIN [OB1] auf.

Der Eingang EN der FC-Box im Codebaustein MAIN muss wahr sein, damit die FC ausgeführt wird.

Der von der Anweisung RET in der FC angegebene Wert liegt nach der Ausführung der FC, an deren Anweisung RET Signalfluss anliegt, am Ausgang ENO der FC-Box im Codebaustein MAIN an.

## 7.8.15 Zykluszeitüberwachung neu starten

Tabelle 7- 115 Anweisung RE\_TRIGR

KOP/FUP	SCL	Beschreibung
	<code>RE_TRIGR ( ) ;</code>	Mit RE_TRIGR (Zykluszeitüberwachung neu starten) kann die maximal zulässige Zeit bis zur Erzeugung einer Fehlermeldung durch die Zykluszeitüberwachung verlängert werden.

Mit der Anweisung RE\_TRIGR starten Sie die Zykluszeitüberwachung während eines Zyklusablaufs neu. Dadurch wird die maximal zulässige Zykluszeit mit der letzten Ausführung der Funktion RE\_TRIGR um eine maximale Zykluszeit verlängert.

### Hinweis

Vor der Firmware-Version 2.2 der S7-1200 CPU musste RE\_TRIGR von einem Programmzyklus-OB ausgeführt werden und konnte dazu verwendet werden, die Zykluszeit des Zielsystems unbegrenzt zu verlängern. ENO = FALSCH und die Zykluszeitüberwachung wird nicht zurückgesetzt, wenn RE\_TRIGR aus einem Anlauf-OB, einem Alarm-OB oder einem Fehler-OB aufgerufen wird.

Ab Firmware-Version 2.2 kann RE\_TRIGR von jedem OB ausgeführt werden (auch Anlauf-, Alarm- und Fehler-OBs). Die Zykluszeit des Zielsystems kann jedoch nur um das 10-Fache der konfigurierten maximalen Zykluszeit verlängert werden.

### Maximale Zykluszeit des PLC-Geräts einstellen

Konfigurieren Sie den Wert für die maximale Zykluszeit in der Gerätekonfiguration unter "Zykluszeit".

Tabelle 7- 116 Zykluszeitwerte

Zykluszeitüberwachung	Mindestwert	Höchstwert	Voreinstellung
Maximale Zykluszeit	1 ms	6000 ms	150 ms

### Laufzeitüberwachung


Endet die maximale Zykluszeit, bevor der Zyklus vollständig ist, wird ein Fehler erzeugt. Wenn ein fehlerverarbeitender Codebaustein OB 80 im Anwenderprogramm enthalten ist, führt die CPU den OB 80 aus, für den Sie Programmlogik eingeben können, um eine bestimmte Reaktion zu erzielen. Ist OB 80 nicht im Programm enthalten, wird die erste Timeout-Bedingung ignoriert und die CPU geht in STOP.

Wenn in demselben Programmzyklus eine zweite Überschreitung der maximalen Zykluszeit auftritt (2 x maximaler Zykluszeitwert), wird ein Fehler ausgelöst, der bewirkt, dass die CPU in den Betriebszustand STOP geht.

Im Betriebszustand STOP wird die Ausführung des Programms angehalten, die CPU-Systemkommunikation und die Systemdiagnose laufen jedoch weiter.

### 7.8.16 Zyklus anhalten

Tabelle 7- 117 Anweisung STP

KOP/FUP	SCL	Beschreibung
	<code>STP ( ) ;</code>	STP (Zyklus stoppen) versetzt die CPU in den Betriebszustand STOP. Ist die CPU im Betriebszustand STOP, so werden die Ausführung des Programms und die Aktualisierung der Ein- und Ausgänge im Prozessbild angehalten.

Weitere Informationen finden Sie unter: Konfigurieren der Ausgänge für den Wechsel von RUN in STOP (Seite 95).

Ist EN = WAHR, so geht die CPU in den Betriebszustand STOP, die Programmausführung endet und der Zustand von ENO ist bedeutungslos. Sonst ist EN = ENO = 0.

### 7.8.17 Operationen Get\_Error

Die Anweisungen GET\_ERROR liefern Informationen über Fehler in der Ausführung von Programmbausteinen. Wenn Sie eine Anweisung GetError oder GetErrorID in Ihren Codebaustein aufnehmen, können Sie Programmfehler in Ihrem Programmbaustein behandeln.



## GetError

Tabelle 7- 118 Anweisung GetError

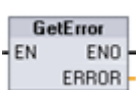
KOP/FUP	SCL	Beschreibung
	<pre>GET_ERROR ( _o ut_ );</pre>	Zeigt an, wenn ein Fehler in der Ausführung eines lokalen Programmbausteins aufgetreten ist und belegt eine vordefinierte Fehlerdatenstruktur mit ausführlichen Fehlerinformationen.

Tabelle 7- 119 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
ERROR	ErrorStruct	Fehlerdatenstruktur: Sie können die Struktur, nicht jedoch die Elemente innerhalb der Struktur umbenennen.

Tabelle 7- 120 Elemente der ErrorStruct-Datenstruktur

Strukturkomponenten	Datentyp	Beschreibung																								
ERROR_ID	Word	Fehler-ID																								
FLAGS	Byte	Zeigt an, ob während eines Bausteinaufrufs ein Fehler aufgetreten ist. <ul style="list-style-type: none"> <li>16#01: Fehler während eines Bausteinaufrufs.</li> <li>16#00: Kein Fehler während eines Bausteinaufrufs.</li> </ul>																								
REACTION	Byte	Standardreaktion: <ul style="list-style-type: none"> <li>0: Ignorieren (Schreibfehler),</li> <li>1: Mit Ersatzwert "0" fortfahren (Lesefehler),</li> <li>2: Anweisung überspringen (Systemfehler)</li> </ul>																								
CODE_ADDRESS	CREF	Informationen zur Adresse und zum Bausteintyp																								
BLOCK_TYPE	Byte	Typ des Bausteins, in dem der Fehler aufgetreten ist: <ul style="list-style-type: none"> <li>1: OB</li> <li>2: FC</li> <li>3: FB</li> </ul>																								
CB_NUMBER	UInt	Nummer des Codebausteins																								
OFFSET	UDInt	Verweis auf den internen Speicher																								
MODE	Byte	Zugriffsart: Je nach Art des Zugriffs können die folgenden Informationen ausgegeben werden: <table border="1" data-bbox="726 1713 1468 1892"> <thead> <tr> <th>Betriebsart</th> <th>(A)</th> <th>(B)</th> <th>(C)</th> <th>(D)</th> <th>(E)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>1</td> <td></td> <td></td> <td></td> <td></td> <td>Offset</td> </tr> <tr> <td>2</td> <td></td> <td></td> <td>Area</td> <td></td> <td></td> </tr> </tbody> </table>	Betriebsart	(A)	(B)	(C)	(D)	(E)	0						1					Offset	2			Area		
Betriebsart	(A)	(B)	(C)	(D)	(E)																					
0																										
1					Offset																					
2			Area																							

Strukturkomponenten		Datentyp	Beschreibung				
			3	Adresse	Umfang		Number (Nummer)
			4			Area	Offset
			5			Area	DB-Nr. Offset
			6	PtrNr./Zugr.		Area	DB-Nr. Offset
			7	PtrNr./Zugr.	Stpl.-Nr./Umfang	Area	DB-Nr. Offset
OPERAND_NUMBER		UInt	Operandennummer des Maschinenbefehls				
POINTER_NUMBER_LOCATION		UInt	(A) Interner Pointer				
SLOT_NUMBER_SCOPE		UInt	(B) Speicherbereich im internen Speicher				
DATA_ADDRESS		NREF	Informationen zur Adresse eines Operanden				
	AREA	Byte	(C) Speicherbereich: <ul style="list-style-type: none"> <li>• L: 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE</li> <li>• E: 16#81</li> <li>• A: 16#82</li> <li>• M: 16#83</li> <li>• DB: 16#84, 85, 8A, 8B</li> </ul>				
	DB_NUMBER	UInt	(D) Nummer des Datenbausteins				
	OFFSET	UDInt	(E) Relative Adresse des Operanden				

### GetErrorID

Tabelle 7- 121 Anweisung GetErrorID


KOP/FUP	SCL	Beschreibung
	<pre>GET_ERR_ID () ;</pre>	Zeigt an, wenn ein Fehler in der Ausführung eines Programmbausteins aufgetreten ist und meldet die ID (Kennung) des Fehlers.

Tabelle 7- 122 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
ID	Word	Fehlerkennwerte für das ErrorStruct-Element ERROR_ID

Tabelle 7- 123 Error\_ID-Werte

ERROR_ID Hexadezimal	ERROR_ID Dezimal	Ausführungsfehler Programmbaustein
0	0	Kein Fehler
2503	9475	Fehler nicht initialisierter Pointer
2522	9506	Lesefehler Operand außerhalb des Bereichs
2523	9507	Schreibfehler Operand außerhalb des Bereichs
2524	9508	Lesefehler ungültiger Bereich
2525	9509	Schreibfehler ungültiger Bereich
2528	9512	Lesefehler Datenausrichtung (fehlerhafte Bitausrichtung)
2529	9513	Schreibfehler Datenausrichtung (fehlerhafte Bitausrichtung)
2530	9520	DB schreibgeschützt
253A	9530	Globaler DB ist nicht vorhanden
253C	9532	Falsche Version oder FC ist nicht vorhanden
253D	9533	Anweisung ist nicht vorhanden
253E	9534	Falsche Version oder FB ist nicht vorhanden
253F	9535	Anweisung ist nicht vorhanden
2575	9589	Fehler in der Schachtelungstiefe des Programms
2576	9590	Zuordnungsfehler Lokaldaten
2942	10562	Physischer Eingang ist nicht vorhanden
2943	10563	Physischer Ausgang ist nicht vorhanden

## Funktionsweise

Standardmäßig reagiert die CPU auf einen Bausteinausführungsfehler durch Eintragen des Fehlers in den Diagnosepuffer. Wenn Sie jedoch eine oder mehrere Anweisungen `GetError` oder `GetErrorID` in einem Codebaustein anordnen, kann dieser Baustein die Fehler im Baustein behandeln. Dann meldet die CPU keinen Fehler im Diagnosepuffer. Stattdessen werden die Fehlerinformationen im Ausgang der Anweisung `GetError` oder `GetErrorID` gemeldet. Ausführliche Fehlerinformationen können Sie mit der Anweisung `GetError` auslesen, mit der Anweisung `GetErrorID` wird nur die Fehlerkennung gelesen. Normalerweise ist der erste Fehler der wichtigste und die weiteren Fehler lediglich Folgen des ersten Fehlers.

Die erste Ausführung einer Anweisung `GetError` oder `GetErrorID` innerhalb eines Bausteins gibt den ersten während der Bausteinausführung erkannten Fehler aus. Dieser Fehler kann an einer beliebigen Stelle zwischen dem Start des Bausteins und der Ausführung von `GetError` oder `GetErrorID` aufgetreten sein. Nachfolgende Ausführungen von `GetError` oder `GetErrorID` geben den ersten Fehler nach der vorherigen Ausführung von `GetError` oder `GetErrorID` aus. Der Verlauf der Fehler wird nicht gespeichert, und die Ausführung der Anweisung initiiert das Zielsystem neu, um den nächsten Fehler zu erfassen.

Der Datentyp `ErrorStruct` der Anweisung `GetError` kann im Datenbausteineditor und in den Bausteinschnittstellen eingefügt werden, damit Ihre Programmlogik auf diese Werte zugreifen kann. Wählen Sie den Datentyp `ErrorStruct` in der Klappliste aus, um diese Struktur zu ergänzen. Mithilfe von eindeutigen Namen können Sie mehrere `ErrorStruct`-Elemente erstellen. Die Elemente eines `ErrorStruct` können nicht umbenannt werden.

### Von ENO angezeigte Fehlerbedingung

Wenn EN = WAHR, und GetError oder GetErrorID wird ausgeführt, dann:

- ENO = WAHR gibt an, dass während der Ausführung eines Codebausteins ein Fehler aufgetreten ist und Fehlerdaten vorliegen
- ENO = FALSCH gibt an, dass während der Ausführung eines Codebausteins kein Fehler aufgetreten ist

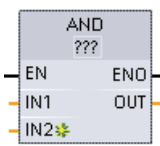
Sie können die Logik des Fehlerverarbeitungsprogramms mit dem Eingang ENO verknüpfen, der nach dem Auftreten eines Fehlers gesetzt wird. Bei einem Fehler werden die Fehlerdaten im Ausgangsparameter gespeichert, wo Ihr Programm darauf zugreifen kann.

Mit GetError und GetErrorID können Fehlerinformationen des aktuellen Bausteins (des aufgerufenen Bausteins) an einen aufrufenden Baustein gesendet werden. Platzieren Sie die Anweisung im letzten Netzwerk des aufgerufenen Bausteinprogramms, um den endgültigen Ausführungsstatus des aufgerufenen Bausteins zu melden.

## 7.9 Wortverknüpfung

### 7.9.1 Operationen AND (UND), OR (ODER) und XOR (EXKLUSIV ODER)

Tabelle 7- 124 Anweisung AND (UND), OR (ODER) und XOR (EXKLUSIV ODER)

KOP/FUP	SCL	Beschreibung
	<code>out := in1 AND in2;</code>	AND: UND-Verknüpfung
	<code>out := in1 OR in2;</code>	OR: ODER-Verknüpfung
	<code>out := in1 XOR in2;</code>	XOR: Exklusiv-ODER-Verknüpfung

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.



Um einen Eingang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".

Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 7- 125 Datentypen für die Parameter


Parameter	Datentyp	Beschreibung
IN1, IN2	Byte, Word, DWord	Verknüpfungseingänge
OUT	Byte, Word, DWord	Verknüpfungsausgang

<sup>1</sup> Durch die Auswahl des Datentyps werden die Parameter IN1, IN2 und OUT auf den gleichen Datentyp gesetzt.

Die entsprechenden Bitwerte von IN1 und IN2 werden zu einem binären Ergebnis in Parameter OUT verknüpft. ENO ist nach der Ausführung dieser Anweisungen immer WAHR.

## 7.9.2 Operation INV (Einerkomplement erzeugen)

Tabelle 7- 126 Anweisung INV

KOP/FUP	SCL	Beschreibung
	Nicht verfügbar	Berechnet das binäre Einerkomplement des Parameters IN. Das Einerkomplement wird gebildet durch Invertieren jedes Bitwerts des Parameters IN (Ändern von 0 in 1 und 1 in 0). ENO ist nach der Ausführung dieser Anweisung immer WAHR.


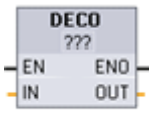
<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 127 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Datenelement für die Invertierung
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Invertierter Ausgangswert

### 7.9.3 Operationen Hexadezimalzahl in Bit wandeln und Bit in Hexadezimalzahl wandeln

Tabelle 7- 128 Anweisungen ENCO und DECO

KOP/FUP	SCL	Beschreibung
	<pre>out := ENCO(_in_);</pre>	<p>Wandelt ein Bitmuster in eine Binärzahl um.</p> <p>Die Anweisung ENCO wandelt den Parameter IN in die Binärzahl um, die der Bitposition des niederwertigsten Bits von Parameter IN entspricht, und gibt das Ergebnis in Parameter OUT aus. Ist Parameter IN entweder 0000 0001 oder 0000 0000, so wird der Wert 0 an Parameter OUT zurückgegeben. Ist der Wert des Parameters IN gleich 0000 0000, dann wird ENO auf FALSCH gesetzt.</p>
	<pre>out := DECO(_in_);</pre>	<p>Wandelt eine Binärzahl in ein Bitmuster um.</p> <p>Die Anweisung DECO wandelt eine Binärzahl aus Parameter IN um, indem die entsprechende Bitposition in Parameter OUT auf 1 gesetzt wird (alle anderen Bits werden auf 0 gesetzt). ENO ist nach der Ausführung der Anweisung DECO immer WAHR.</p> <p>Hinweis: Der Standarddatentyp für die Anweisung DECO ist DWORD. In SCL ändern Sie den Anweisungsnamen in DECO_BYTE oder DECO_WORD, um einen Byte- oder Wortwert zu decodieren, und Sie weisen eine Byte- oder Wortvariable oder -adresse zu.</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 7- 129 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	ENCO: Byte, Word, DWord DECO: UInt	ENCO: Bitmuster für die Umwandlung DECO: Binärwert für die Umwandlung
OUT	ENCO: Int DECO: Byte, Word, DWord	ENCO: Umgewandelter Wert DECO: Umgewandeltes Bitmuster

Tabelle 7- 130 OUT-Parameter für ENCO

ENO	Bedingung	Ergebnis (OUT)
1	Kein Fehler	Gültige Bitnummer
0	IN ist Null	OUT wird auf Null gesetzt

Die möglichen Datentypen Byte, Word oder DWord für den DECO-Parameter OUT begrenzen den nützlichen Bereich des Parameters IN. Wenn der Wert des Parameters IN den nützlichen Bereich überschreitet, wird eine Modulo-Operation durchgeführt, um die niederwertigsten Bits zu extrahieren (siehe unten).

Bereich für den DECO-Parameter IN:

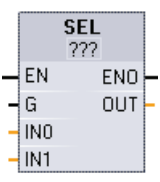
- 3 Bits (Werte 0-7) IN werden verwendet, um 1 Bitposition in einem Byte OUT zu setzen
- 4 Bits (Werte 0-15) IN werden verwendet, um 1 Bitposition in einem Word OUT zu setzen
- 5 Bits (Werte 0-31) IN werden verwendet, um 1 Bitposition in einem DWord OUT zu setzen

Tabelle 7- 131 Beispiele

Wert IN für DECO			Wert OUT für DECO (Einzelne Bitposition umwandeln)
Byte OUT 8 Bits	Min. IN	0	00000001
	Max. IN	7	10000000
Word OUT 16 Bits	Min. IN	0	0000000000000001
	Max. IN	15	1000000000000000
DWord OUT 32 Bits	Min. IN	0	00000000000000000000000000000001
	Max. IN	31	10000000000000000000000000000000

## 7.9.4 Anweisungen Selektieren, Multiplexen und Demultiplexen

Tabelle 7- 132 Anweisung SEL (Selektieren)

KOP/FUP	SCL	Beschreibung
	<pre>out := SEL(   g:=_bool_in,   in0:=_variant_in,   in1:=_variant_in);</pre>	<p>Die Anweisung SEL weist einen von zwei Eingangswerten, abhängig vom Wert des Parameters G, dem Parameter OUT zu.</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

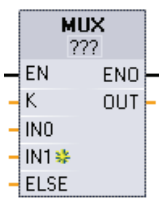
Tabelle 7- 133 Datentypen für die Anweisung SEL

Parameter	Datentyp <sup>1</sup>	Beschreibung
G	Bool	<ul style="list-style-type: none"> <li>• 0 wählt IN0</li> <li>• 1 wählt IN1</li> </ul>
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Eingänge
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Ausgang

<sup>1</sup> Die Eingangsvariablen und die Ausgangsvariable müssen vom gleichen Datentyp sein.

**Bedingungscode:** ENO ist nach der Ausführung der Anweisung SEL immer WAHR.

Tabelle 7- 134 Anweisung MUX (Multiplexen)

KOP/FUP	SCL	Beschreibung
	<pre> out := MUX(     k:=_unit_in,     in1:=variant_in,     in2:=variant_in,      [...in32:=variant_in,]     in_else:=variant_in);                     </pre>	<p>Die Anweisung MUX kopiert einen von vielen Eingangswerten, abhängig vom Wert des Parameters K, in den Parameter OUT. Überschreitet der Wert von Parameter K (<math>IN_n - 1</math>), so wird der Wert des Parameters ELSE in den Parameter OUT kopiert.</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.



Um einen Eingang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".

Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

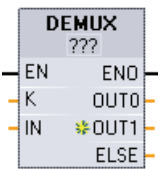
Tabelle 7- 135 Datentypen für die Anweisung MUX

Parameter	Datentyp	Beschreibung
K	UInt	<ul style="list-style-type: none"> <li>• 0 wählt IN1</li> <li>• 1 wählt IN2</li> <li>• <math>n</math> wählt <math>IN_n</math></li> </ul>
IN0, IN1, .. INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Eingänge
ELSE	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Eingang Ersatzwert (optional)
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	Ausgang

<sup>1</sup> Die Eingangsvariablen und die Ausgangsvariable müssen vom gleichen Datentyp sein.



Tabelle 7- 136 Anweisung DEMUX (Demultiplexen)

KOP/FUP	SCL	Beschreibung
	<pre> DEMUX (     k:=_unit_in,     in:=variant_in,     out1:=variant_in,     out2:=variant_in,      [...out32:=variant_in,]     outelse:=variant_in); </pre>	<p>DEMUX kopiert den Wert der Adresse, die dem Parameter IN zugewiesen ist, in einen von vielen Ausgängen. Der Wert des Parameters K gibt an, welcher Ausgang als Ziel des Werts IN ausgewählt wird. Ist der Wert von K größer als die Zahl (OUT<sub>n</sub> - 1), dann wird der Wert IN in die Adresse kopiert, die dem Parameter ELSE zugewiesen ist.</p>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Um einen Ausgang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss und wählen den Befehl "Ausgang einfügen". Um einen Ausgang zu löschen, klicken Sie bei einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss (sofern mehr als die zwei ursprünglichen Ausgänge vorhanden sind) und wählen den Befehl "Löschen".



Um einen Ausgang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss und wählen den Befehl "Ausgang einfügen".

Um einen Ausgang zu löschen, klicken Sie bei einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss (sofern mehr als die zwei ursprünglichen Ausgänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 7- 137 Datentypen für die Anweisung DEMUX

Parameter	Datentyp <sup>1</sup>	Beschreibung
K	UInt	Auswahlwert: <ul style="list-style-type: none"> <li>• 0 wählt OUT1</li> <li>• 1 wählt OUT2</li> <li>• n wählt OUTn</li> </ul>
IN	SInt, Int, DInt, USInt, UInt, UInt, Real, LReal, Byte, Word, DWord, Time, Char	Eingang
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UInt, Real, LReal, Byte, Word, DWord, Time, Char	Ausgänge
ELSE	SInt, Int, DInt, USInt, UInt, UInt, Real, LReal, Byte, Word, DWord, Time, Char	Ausgang ersetzen, wenn K größer als (OUTn - 1) ist

<sup>1</sup> Die Eingangsvariable und die Ausgangsvariablen müssen vom gleichen Datentyp sein.

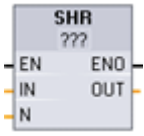
Tabelle 7- 138 ENO-Zustand für die Anweisungen MUX und DEMUX

ENO	Bedingung	Ergebnis OUT
1	Kein Fehler	MUX: Ausgewählter IN-Wert wird in OUT kopiert DEMUX: IN-Wert wird in ausgewählten OUT kopiert
0	MUX: K ist größer als die Anzahl der Eingänge - 1	<ul style="list-style-type: none"> <li>Kein ELSE angegeben: OUT wird nicht verändert,</li> <li>ELSE angegeben, ELSE-Wert wird OUT zugewiesen</li> </ul>
	DEMUX: K ist größer als die Anzahl der Ausgänge -1	<ul style="list-style-type: none"> <li>Kein ELSE angegeben: Ausgänge werden nicht verändert,</li> <li>ELSE angegeben, IN-Wert wird in ELSE kopiert</li> </ul>

## 7.10 Schieben und Rotieren

### 7.10.1 Schiebeoperationen

Tabelle 7- 139 Anweisungen SHR und SHL

KOP/FUP	SCL	Beschreibung
	<pre> out := SHR(     in:=_variant_in_,     n:=_uint_in_); out := SHL(     in:=_variant_in_,     n:=_uint_in_);                     </pre>	Mit den Schiebeanweisungen (SHL und SHR) schieben Sie das Bitmuster von Parameter IN. Das Ergebnis wird dem Parameter OUT zugewiesen. Parameter N gibt die Anzahl der geschobenen Bitpositionen an: <ul style="list-style-type: none"> <li>SHR: Bitmuster nach rechts schieben</li> <li>SHL: Bitmuster nach links schieben</li> </ul>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.

Tabelle 7- 140 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Byte, Word, DWord	Bitmuster, das verschoben werden soll
N	UInt	Anzahl der Bitpositionen, die verschoben werden sollen
OUT	Byte, Word, DWord	Bitmuster nach dem Schieben

- Bei N=0 wird nicht geschoben. Der Wert von IN wird OUT zugewiesen.
- In die Bitpositionen, die von der Schiebeanweisung geleert wurden, werden Nullen geschoben.

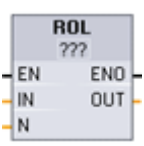
- Falls die Anzahl der zu schiebenden Positionen (N) die Anzahl der Bits im Zielwert überschreitet (8 bei Byte, 16 bei Word, 32 bei DWord), dann werden alle ursprünglichen Bitwerte hinausgeschoben und durch Nullen ersetzt (OUT wird Null zugewiesen).
- ENO ist bei den Schiebeanweisungen immer WAHR.

Tabelle 7- 141 Beispiel für SHL mit Word-Daten

Schiebt die Bits eines Worts nach links, indem von rechts Nullen eingefügt werden (N = 1)			
IN	1110 0010 1010 1101	Wert von OUT vor dem ersten Schieben:	1110 0010 1010 1101
		Nach dem ersten Schieben links:	1100 0101 0101 1010
		Nach dem zweiten Schieben links:	1000 1010 1011 0100
		Nach dem dritten Schieben links:	0001 0101 0110 1000

## 7.10.2 Rotieroperationen

Tabelle 7- 142 Anweisungen ROR und ROL

KOP/FUP	SCL	Beschreibung
	<pre> out := ROL(     in:=_variant_in_,     n:=_uint_in_); out := ROR(     in:=_variant_in_,     n:=_uint_in_); </pre>	<p>Mit den Rotieranweisungen (ROR und ROL) können Sie das Bitmuster von Parameter IN rotieren. Das Ergebnis wird Parameter OUT zugewiesen. Parameter N legt die Zahl der rotierten Bitpositionen fest.</p> <ul style="list-style-type: none"> <li>• ROR: Bitmuster nach rechts rotieren</li> <li>• ROL: Bitmuster nach links rotieren</li> </ul>

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.

Tabelle 7- 143 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Byte, Word, DWord	Bitmuster, das rotiert werden soll
N	UInt	Anzahl der Bitpositionen, die rotiert werden sollen
OUT	Byte, Word, DWord	Bitmuster nach dem Rotieren

- Bei N=0 wird nicht rotiert. Der Wert von IN wird OUT zugewiesen.
- Bitdaten, die auf einer Seite des Zielwerts hinausrotiert werden, werden auf der anderen Seite des Zielwerts hineinrotiert, so dass keine ursprünglichen Bitwerte verloren gehen.
- Falls die Anzahl der zu rotierenden Bitpositionen (N) die Anzahl der Bits im Zielwert überschreitet (8 bei Byte, 16 bei Word, 32 bei DWord), dann wird trotzdem rotiert.
- ENO ist nach der Ausführung der Rotieranweisungen immer WAHR.

Tabelle 7- 144 Beispiel für ROR mit Word-Daten

<b>Bits rechts hinausrotieren und links hineinrotieren (N = 1)</b>			
IN	0100 0000 0000 0001	Wert von OUT vor dem ersten Rotieren:	0100 0000 0000 0001
		Nach dem ersten Rotieren rechts:	1010 0000 0000 0000
		Nach dem zweiten Rotieren rechts:	0101 0000 0000 0000

## Erweiterte Anweisungen

### 8.1 Datum und Uhrzeit

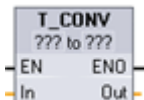
#### 8.1.1 Datum- und Uhrzeitoperationen

Mit den Datum- und Uhrzeitanweisungen können Sie Datums- und Zeitberechnungen programmieren.

- T\_CONV wandelt den Datentyp eines Zeitwerts um: (Time in DInt) oder (DInt in Time).
- T\_ADD addiert Time- und DTL-Werte: (Time + Time = Time) oder (DTL + Time = DTL)
- T\_SUB subtrahiert Time- und DTL-Werte: (Time - Time = Time) oder (DTL - Time = DTL)
- T\_DIFF gibt die Differenz zwischen zwei DTL-Werten als Time-Wert aus: DTL - DTL = Time
- T\_COMBINE verknüpft einen Date-Wert und einen Time\_and\_Date-Wert, um einen DTL-Wert zu erhalten

Informationen zur Struktur von DTL- und Time-Daten finden Sie im Abschnitt zu den Uhrzeit- und Datums-Datentypen (Seite 105).

Tabelle 8- 1 Anweisung T\_CONV (Zeiten umwandeln)


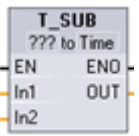
KOP/FUP	SCL	Beschreibung
	<pre>out := T_CONV( in:=_variant_in);</pre>	T_CONV wandelt den Datentyp Time in den Datentyp DInt oder umgekehrt den Datentyp DInt in den Datentyp Time um.

<sup>1</sup> In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.

Tabelle 8- 2 Datentypen für die Parameter von T\_CONV

Parameter und Datentyp	Datentyp	Beschreibung
IN	DInt, Time	Time-Eingabewert oder DInt-Wert
OUT	DInt, Time	Konvertierter DInt-Wert oder Time-Wert

Tabelle 8-3 Anweisungen T\_ADD (Zeiten addieren) und T\_SUB (Zeiten subtrahieren)

KOP/FUP	SCL	Beschreibung
	<pre>out := T_ADD(   in1:=_variant_in,   in2:=_time_in);</pre>	<p>T_ADD addiert den Wert aus Eingang IN1 (Datentyp DTL oder Time) zum Wert in Eingang IN2 (Datentyp Time). Der Parameter OUT liefert das Ergebnis als Wert vom Datentyp DTL oder Time. Zwei Datentyp-Anweisungen sind möglich:</p> <ul style="list-style-type: none"> <li>• Time + Time = Time</li> <li>• DTL + Time = DTL</li> </ul>
	<pre>out := T_SUB(   in1:=_variant_in,   in2:=_time_in);</pre>	<p>T_SUB subtrahiert den Wert aus Eingang IN2 (Datentyp Time) vom Wert von IN1 (Datentyp DTL oder Time). Der Parameter OUT liefert den Differenzwert als Datentyp DTL oder Time. Zwei Datentyp-Anweisungen sind möglich:</p> <ul style="list-style-type: none"> <li>• Time - Time = Time</li> <li>• DTL - Time = DTL</li> </ul>

1 In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.

Tabelle 8-4 Datentypen für die Parameter von T\_ADD und T\_SUB

Parameter und Datentyp	Datentyp	Beschreibung
IN1 <sup>1</sup>	IN	DTL, Time
IN2	IN	Time
OUT	OUT	DTL, Time

1 Wählen Sie den Datentyp für IN1 aus der Klappliste unter dem Namen der Anweisung aus. Durch die Auswahl des Datentyps für IN1 wird auch der Datentyp für den Parameter OUT gesetzt.

Tabelle 8-5 Anweisung T\_DIFF (Zeitdifferenz)


KOP/FUP	SCL	Beschreibung
	<pre>out := T_DIFF(   in1:=_DTL_in,   in2:=_DTL_in);</pre>	<p>T_DIFF subtrahiert den DTL-Wert (IN2) vom DTL-Wert (IN1). Der Parameter OUT liefert den Differenzwert als Datentyp Time.</p> <ul style="list-style-type: none"> <li>• DTL - DTL = Time</li> </ul>

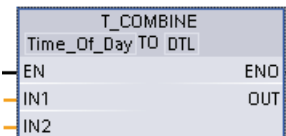
Tabelle 8-6 Datentypen für die Parameter von T\_DIFF

Parameter und Datentyp	Datentyp	Beschreibung
IN1	IN	DTL
IN2	IN	DTL
OUT	OUT	Time

**Bedingungscode:** ENO = 1 bedeutet, dass kein Fehler aufgetreten ist. ENO = 0 und Parameter OUT = 0 Fehler:

- Ungültiger DTL-Wert
- Ungültiger Time-Wert

Tabelle 8-7 Anweisung T\_COMBINE (Zeitwerte verknüpfen)

KOP/FUP	SCL	Beschreibung
	<pre>out := CONCAT_DATE_TOD(   In1 := _date_in,   In2 := _tod_in);</pre>	T_COMBINE verknüpft einen Date-Wert und einen Time_of_Day-Wert, um einen DTL-Wert zu erhalten.

<sup>1</sup> Beachten Sie, dass die Anweisung T\_COMBINE in den erweiterten Anweisungen der Funktion CONCAT\_DATE\_TOD in SCL entspricht.

Tabelle 8-8 Datentypen für die Parameter von T\_COMBINE

Parameter und Datentyp		Datentyp	Beschreibung
IN1	IN	Date	Der zu verknüpfende Date-Wert muss zwischen DATE#1990-01-01 und DATE#2089-12-31 liegen
IN2	IN	Time_of_Day	Zu verknüpfende Time_of_Day-Werte
OUT	OUT	DTL	DTL-Wert

### 8.1.2 Systemuhr einstellen und lesen

Mit den Uhrzeitanweisungen können Sie die Systemuhr der CPU einstellen und lesen. Der Datentyp DTL (Seite 105) stellt Datum- und Uhrzeitwerte bereit.

Tabelle 8-9 Anweisungen für die Systemzeit


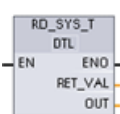
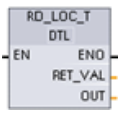
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := WR_SYS_T(   in:=_DTL_in);</pre>	WR_SYS_T (Systemzeit schreiben) stellt die Echtzeituhr in der CPU anhand des DTL-Werts von Parameter IN ein. Dieser Zeitwert berücksichtigt weder die jeweilige Ortszeit noch die Sommer-/Winterzeitschaltung.
	<pre>ret_val := RD_SYS_T(   out=&gt;_DTL_out);</pre>	Mit RD_SYS_T (Systemzeit lesen) wird die aktuelle Systemzeit der CPU ausgelesen. Dieser Zeitwert berücksichtigt weder die jeweilige Ortszeit noch die Sommer-/Winterzeitschaltung.
	<pre>ret_val := RD_LOC_T(   out=&gt;_DTL_out);</pre>	RD_LOC_T (Lokalzeit lesen) stellt die aktuelle Lokalzeit der CPU als Datentyp DTL bereit. Dieser Uhrzeitwert entspricht der lokalen Zeitzone sowie der Anpassung an Sommer-/Winterzeit (sofern konfiguriert).

Tabelle 8- 10 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	DTL	Einzustellende Uhrzeit für die Systemuhr der CPU
RET_VAL	OUT	Int	Ausführungsbedingung
OUT	OUT	DTL	RD_SYS_T: Aktuelle Systemzeit der CPU RD_LOC_T: Aktuelle Lokalzeit, einschließlich Anpassung an Sommer-/Winterzeit (sofern konfiguriert)

- Zur Berechnung der Lokalzeit werden die Zeitzone und die Sommer-/Winterzeitumschaltungen, die Sie im Register "Allgemein" der Gerätekonfiguration unter "Uhrzeit" eingegeben haben, verwendet.
- Die Zeitzone ist bezogen auf die UTC- oder GMT-Systemzeit.
- Für die Sommerzeitumschaltung sind Monat, Woche, Tag und Stunde der Umstellung auf Sommerzeit einzugeben.
- Für die Winterzeitumschaltung sind ebenfalls Monat, Woche, Tag und Stunde der Umstellung auf Winterzeit einzugeben.
- Die Zeitzonendifferenz gilt immer bezogen auf die Systemzeit. Die Sommerzeitverschiebung gilt nur, wenn eine Umstellung auf Sommerzeit stattfindet.

**Hinweis**

**Anfangszeit von Sommer- und Winterzeit einstellen**

Für die Eigenschaften "Uhrzeit" unter "Anfangszeit von Sommer- und Winterzeit" in der CPU-Gerätekonfiguration müssen Sie Ihre lokale Uhrzeit angeben.

**Bedingungscode:** ENO = 1 bedeutet, dass kein Fehler aufgetreten ist. ENO = 0 bedeutet, dass ein Ausführungsfehler aufgetreten ist und ein Bedingungscode am Ausgang RET\_VAL ansteht.

Tabelle 8- 11 Bedingungscode

RET_VAL (W#16#....)	Beschreibung
0000	Die aktuelle Lokalzeit wird als Winterzeit angegeben.
0001	Es wurde die Sommerzeit wurde konfiguriert und die aktuelle Lokalzeit wird als Sommerzeit angegeben.
8080	Keine Ortszeit vorhanden
8081	Jahr ungültig
8082	Monat ungültig
8083	Tag ungültig
8084	Angabe der Stunden ungültig
8085	Angabe der Minuten ungültig
8086	Angabe der Sekunden ungültig
8087	Angabe der Nanosekunden ungültig
80B0	Die Echtzeituhr ist ausgefallen.



### 8.1.3 Betriebsstundenzähler

Tabelle 8- 12 Anweisung RTM

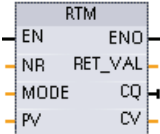
KOP/FUP	SCL	Beschreibung
	<pre>RTM(NR:=_uint_in_,     MODE:=_byte_in_,     PV:=_dint_in_,     CQ=&gt;_bool_out_,     CV=&gt;_dint_out_);</pre>	Die Anweisung RTM (Betriebsstundenzähler) kann die Betriebsstundenzähler in der CPU setzen, starten, stoppen und lesen.

Tabelle 8- 13 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
NR	IN	UInt	Nummer des Betriebsstundenzählers: (mögliche Werte: 0..9)
MODE	IN	Byte	Nummer des RTM-Ausführungsmodus: <ul style="list-style-type: none"> <li>• 0 = Werte abrufen (der Zustand wird dann in CQ und der aktuelle Wert in CV geschrieben)</li> <li>• 1 = Starten (ab dem letzten Zählerwert)</li> <li>• 2 = Stoppen</li> <li>• 4 = Setzen (auf den in PV angegebenen Wert)</li> <li>• 5 = Setzen (auf den in PV angegebenen Wert) und dann starten</li> <li>• 6 = Setzen (auf den in PV angegebenen Wert) und dann stoppen</li> <li>• 7 = Alle RTM-Werte der CPU auf der MC (Memory Card) speichern</li> </ul>
PV	IN	DInt	Stundenwert für den angegebenen Betriebsstundenzähler voreinstellen
RET_VAL	OUT	Int	Funktionsergebnis / Fehlermeldung
CQ	OUT	Bool	Zustand Betriebsstundenzähler (1 = läuft)
CV	OUT	DInt	Aktueller Stundenwert des angegebenen Betriebsstundenzählers

Die CPU betreibt bis zu 10 Betriebsstundenzähler, um die Betriebsstunden von kritischen Untersystemen der Steuerung zu verfolgen. Sie müssen die einzelnen Betriebsstundenzähler mit einer RTM-Anweisung je Zähler starten. Wenn die CPU von RUN in STOP wechselt, werden alle Betriebsstundenzähler gestoppt. Einzelne Zähler können Sie auch mit Ausführungsmodus 2 der RTM-Anweisung stoppen.

Wenn eine CPU von STOP in RUN wechselt, müssen Sie die Betriebsstundenzähler mit einer RTM-Anweisung je Zähler neu starten. Ist der Wert eines Betriebsstundenzählers größer als 2.147.483.647 Stunden, wird der Zählbetrieb beendet und ein Überlauffehler gemeldet. Um einen Zähler zurückzusetzen oder zu ändern, müssen Sie die Anweisung RTM einmal pro Zähler ausführen.

Ein Spannungsausfall oder Neustart der CPU bewirkt einen Abschaltvorgang, bei dem die aktuellen Werte der Betriebsstundenzähler im remanenten Speicher abgelegt werden. Beim Anlauf der CPU werden die gespeicherten Werte der Betriebsstundenzähler erneut in die Zähler geladen, so dass die vorherigen Betriebsstunden nicht verloren sind. Die Betriebsstundenzähler müssen neu gestartet werden, um weitere Betriebsstunden zu zählen.

Ihr Programm kann zudem über Ausführungsmodus 7 der RTM-Anweisung die Werte der Betriebsstundenzähler auf einer Memory Card speichern. Zum Zeitpunkt der Ausführung von Modus 7 der RTM-Anweisung werden die Zustände aller Betriebsstundenzähler auf der Memory Card gespeichert. Diese gespeicherten Werte können im Lauf der Zeit durch Start- und Stoppvorgänge der Betriebsstundenzähler im Programmverlauf inkorrekt werden. Sie müssen die Werte auf der Memory Card regelmäßig aktualisieren, um wichtige Laufzeitergebnisse zu erfassen. Der Vorteil dabei, die RTM-Werte auf der Memory Card zu speichern, ist der, dass Sie die Memory Card in eine Ersatz-CPU einschieben können, wo Ihr Programm und die gespeicherten RTM-Werte dann zur Verfügung stehen. Wenn Sie die RTM-Werte nicht auf einer Memory Card gespeichert haben, sind die Betriebsstundenwerte verloren (bei Nutzung einer Ersatz-CPU).

---

**Hinweis**

**Vermeiden Sie übermäßige Programmaufrufe bei Schreibvorgängen auf der Memory Card**

Halten Sie die Anzahl der Schreibvorgänge auf der Memory Card möglichst gering, um die Lebensdauer der Memory Card zu verlängern.

---

Tabelle 8- 14 Bedingungscode

RET_VAL (W#16#....)	Beschreibung
0	Kein Fehler
8080	Fehlerhafte Nummer des Betriebsstundenzählers
8081	Ein negativer Wert wurde an den Parameter PV übergeben
8082	Überlauf des Betriebsstundenzählers
8091	Der Eingangsparameter MODE enthält einen unzulässigen Wert.
80B1	Wert kann nicht auf MC gespeichert werden (MODUS = 7)

## 8.1.4 Anweisung SET\_TIMEZONE

Tabelle 8- 15 Anweisung SET\_TIMEZONE

KOP/FUP	SCL	Beschreibung
	<pre>"SET_TIMEZONE_DB" (     REQ:=_bool_in,     Timezone:=_struct_in,     DONE=&gt;_bool_out_,     BUSY=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	Stellt die Parameter für lokale Zeitzone und Sommer-/Winterzeitumschaltung ein, anhand deren die CPU-Systemzeit in die jeweilige Ortszeit umgewandelt wird.

<sup>1</sup> Im SCL-Beispiel ist "SET\_TIMEZONE\_DB" der Name des Instanz-DBs.

Tabelle 8- 16 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
Timezone	IN	TimeTransformationRule
DONE	OUT	Bool
BUSY	OUT	Bool
ERROR	OUT	Bool
STATUS	OUT	Word

Um die Zeitzonenparameter für die CPU manuell zu konfigurieren, verwenden Sie die Eigenschaften "Uhrzeit" im Register "Allgemein" der Gerätekonfiguration.

Mit der Anweisung SET\_TIMEZONE stellen Sie die Ortszeit programmatisch ein. Die Parameter der Struktur "TimeTransformationRule" geben die lokale Zeitzone sowie die Zeiteinstellungen für die automatische Umschaltung zwischen Sommer- und Winterzeit an.

Tabelle 8- 17 "Struktur "TimeTransformationRule"

Parameter	Datentyp	Beschreibung
Bias	Int	Zeitunterschied zwischen UTC und Ortszeit [min]
DaylightBias	Int	Zeitunterschied zwischen Winter- und Sommerzeit [min]
DaylightStartMonth	USInt	Monat der Sommerzeit
DaylightStartWeek	USInt	Woche der Sommerzeit: <ul style="list-style-type: none"> <li>• 1 = Erstes Auftreten des Wochentags im Monat</li> <li>• ...</li> <li>• 5 = Letztes Auftreten des Wochentags im Monat</li> </ul>

Parameter	Datentyp	Beschreibung
DaylightStartWeekday	USInt	Wochentag der Sommerzeit: <ul style="list-style-type: none"> <li>• 1 = Sonntag</li> <li>• ...</li> <li>• 7 = Samstag</li> </ul>
DaylightStartHour	USInt	Stunde der Sommerzeit
StandardStartMonth	USInt	Monat der Umschaltung zur Winterzeit
StandardStartWeek	USInt	Woche der Umschaltung zur Winterzeit <ul style="list-style-type: none"> <li>• 1 = Erstes Auftreten des Wochentags im Monat</li> <li>• ...</li> <li>• 5 = Letztes Auftreten des Wochentags im Monat</li> </ul>
StandardStartWeekday	USInt	Wochentag der Winterzeit: <ul style="list-style-type: none"> <li>• 1 = Sonntag</li> <li>• ...</li> <li>• 7 = Samstag</li> </ul>
StandardStartHour	USInt	Stunde der Winterzeit
Time Zone Name	STRING [80]	Name der Zeitzone: (GMT +01:00) Amsterdam, Berlin, Bern, Rom, Stockholm, Wien

## 8.2 Zeichenketten- und Zeichenanweisungen

### 8.2.1 Datentyp String

#### Datentyp STRING

Daten des Typs String werden mit einem Header aus 2 Bytes gefolgt von bis zu 254 Zeichenbytes im ASCII-Code gespeichert. Ein Header für String-Daten hat zwei Längenbytes. Das erste Byte enthält die maximale Länge, die bei der Initialisierung einer Zeichenkette in eckigen Klammern angegeben bzw. auf 254 voreingestellt ist. Das zweite Header-Byte enthält die tatsächliche Länge, die der Anzahl gültiger Zeichen in der Zeichenkette entspricht. Die tatsächliche Länge muss kleiner oder gleich der maximalen Länge sein. Die Anzahl gespeicherter Bytes für das String-Format ist um 2 Bytes größer als die maximale Länge.

#### String-Daten initialisieren

Die Ein- und Ausgangsdaten des Datentyps String müssen vor der Ausführung von Anweisungen als gültige Zeichenketten im Speicher initialisiert werden.

## Gültige String-Daten

Gültige String-Daten haben eine maximale Länge, die größer als null und kleiner als 255 sein muss. Die tatsächliche Länge muss kleiner oder gleich der maximalen Länge sein.

String-Daten können den Speicherbereichen E oder A nicht zugewiesen werden.

Weitere Informationen finden Sie unter: Format des Datentyps String (Seite 106).

### 8.2.2 Anweisung S\_MOVE

Tabelle 8- 18 Anweisung Zeichenkette verschieben

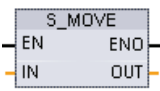
KOP/FUP	SCL	Beschreibung
	<code>out := in;</code>	Zeichenkette der Quelle IN in die Adresse OUT kopieren. Die Ausführung S_MOVE wirkt sich nicht auf den Inhalt der Quellzeichenkette aus.

Tabelle 8- 19 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	String	Quellzeichenkette
OUT	String	Zieladresse

Wenn die tatsächliche Länge der Zeichenkette am Eingang IN die maximale Länge einer am Ausgang OUT gespeicherten Zeichenkette überschreitet, dann wird der Teil der Zeichenkette IN, der in OUT passt, kopiert.

## 8.2.3 Zeichenkettenkonvertierungsoperationen

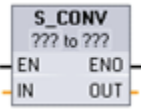
### 8.2.3.1 Konvertierung Zeichenketten in Werte und Werte in Zeichenketten

Mit den folgenden Anweisungen können Sie Zahlenzeichenketten in Zahlenwerte und Zahlenwerte in Zahlenzeichenketten umwandeln:

- Mit S\_CONV wird eine Zahlenzeichenkette in einen Zahlenwert oder ein Zahlenwert in eine Zahlenzeichenkette umgewandelt.
- Mit STRG\_VAL wird eine Zahlenzeichenkette in einen Zahlenwert mit Formatoptionen umgewandelt.
- Mit VAL\_STRG wird ein Zahlenwert in eine Zahlenzeichenkette mit Formatoptionen umgewandelt.

### S\_CONV (Umwandlung von Zeichenkette in Wert)

Tabelle 8- 20 Anweisungen für die Zeichenkettenkonvertierung

KOP/FUP	SCL	Beschreibung
	<pre>out := &lt;Type&gt;_TO_&lt;Type&gt;(in) ;</pre>	Konvertiert eine Zeichenkette in den entsprechenden Wert bzw. einen Wert in die entsprechende Zeichenkette. Die Anweisung S_CONV verfügt über keine Optionen zur Formatierung des Ausgangswerts. Daher ist S_CONV einfacher, aber auch weniger flexibel als die Anweisungen STRG_VAL und VAL_STRG.

- 1 In KOP/FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.
- 2 In SCL: Wählen Sie S\_CONV aus den erweiterten Anweisungen und beantworten Sie die Abfragen der Datentypen für die Umwandlung. STEP 7 stellt daraufhin die entsprechende Umwandlungsanweisung zur Verfügung.

Tabelle 8- 21 Datentypen (Zeichenkette in Zahlenwert wandeln)

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	String	Eingegebene Zeichenkette
OUT	OUT	String, Char, SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Ausgegebener Zahlenwert

Die Umwandlung des Zeichenkettenparameters IN beginnt mit dem ersten Zeichen und wird bis zum Ende der Zeichenkette oder bis zum ersten Zeichen, das nicht "0" bis "9", "+", "-" oder "." ist, fortgesetzt. Das Ergebnis steht an der mit Parameter OUT angegebenen Adresse bereit. Passt der ausgegebene Zahlenwert nicht in den Bereich des Datentyps von OUT, so wird der Parameter OUT auf 0 und ENO auf FALSCH gesetzt. Andernfalls enthält der Parameter OUT ein gültiges Ergebnis und ENO wird auf WAHR gesetzt.

Formatregeln für die Zeichenketteneingabe:

- Enthält die Zeichenkette in IN ein Dezimalzeichen, so muss das Zeichen "." verwendet werden.
- Kommazeichen "," als Tausendertrennzeichen links vom Dezimalpunkt sind zulässig, werden aber ignoriert.
- Führende Leerzeichen werden ignoriert.

### S\_CONV (Umwandlung von Wert in Zeichenkette)

Tabelle 8- 22 Datentypen (Zahlenwert in Zeichenkette wandeln)

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	String, Char, SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Eingegebener Zahlenwert
OUT	OUT	String	Ausgegebene Zeichenkette

Ein ganzzahliger Wert mit oder ohne Vorzeichen oder ein Gleitpunktwert IN wird in die entsprechende Zeichenkette umgewandelt und steht in Parameter OUT bereit. Der Parameter OUT muss eine gültige Zeichenkette referenzieren, bevor die Umwandlung erfolgt. Eine gültige Zeichenkette besteht aus der maximalen Zeichenkettenlänge im ersten Byte, der tatsächlichen Zeichenkettenlänge im zweiten Byte und den tatsächlichen Zeichen in den nächsten Bytes. Die umgewandelte Zeichenkette ersetzt Zeichen in der Zeichenkette OUT, beginnend mit dem ersten Zeichen, und passt das Byte mit der tatsächlichen Länge der ausgegebenen Zeichenkette entsprechend an. Das Byte mit der maximalen Länge der ausgegebenen Zeichenkette wird nicht verändert.

Wie viele Zeichen ersetzt werden, hängt von Datentyp und Zahlenwert des Parameters IN ab. Die ersetzte Zeichenzahl muss in die in Parameter OUT festgelegte Zeichenkettenlänge passen. Die maximale Zeichenkettenlänge (erstes Byte) der Zeichenkette in Parameter OUT muss größer oder gleich der maximal erwarteten Zahl umgewandelter Zeichen sein. Die folgende Tabelle zeigt die maximal möglichen Zeichenkettenlängen, die für jeden unterstützten Datentyp erforderlich sind.

Tabelle 8- 23 Maximale Zeichenkettenlänge für jeden Datentyp


Datentyp IN	Maximale Anzahl umgewandelter Zeichen in der Zeichenkette OUT	Beispiel	Gesamtlänge der Zeichenkette mit den Bytes für maximale und tatsächliche Länge
USInt	3	255	5
SInt	4	-128	6
UInt	5	65535	7
Int	6	-32768	8
UDInt	10	4294967295	12
DInt	11	-2147483648	13

Formatregeln für die Zeichenkettenausgabe:

- In Parameter OUT geschriebene Werte verwenden kein führendes "+" Zeichen.
- Es wird die Festpunktdarstellung verwendet (keine Exponentialdarstellung).
- Mit dem Punktzeichen "." wird der Dezimalpunkt dargestellt, wenn der Parameter IN vom Datentyp Real ist.

## Anweisung STRG\_VAL

Tabelle 8- 24 Anweisung Zeichenkette in Zahlenwert wandeln

KOP/FUP	SCL	Beschreibung
	<pre>"STRG_VAL" (     in:=_string_in,     format:=_word_in,     p:=uint_in,     out=&gt;_variant_out);</pre>	Konvertiert eine Zahlenzeichenkette in die entsprechende ganzzahlige oder Gleitpunktdarstellung.

<sup>1</sup> In KOP/FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.






Regeln für die STRG\_VAL-Umwandlung:

- Wird das Punktzeichen "." für den Dezimalpunkt benutzt, so werden Kommazeichen "," links vom Dezimalpunkt als Tausendertrennzeichen interpretiert. Kommazeichen sind zulässig und werden ignoriert.
- Wird das Kommazeichen "," als Dezimaltrennzeichen benutzt, so werden Punktzeichen "." links vom Dezimalzeichen als Tausendertrennzeichen interpretiert. Diese Punktzeichen sind zulässig und werden ignoriert.
- Führende Leerzeichen werden ignoriert.

## Anweisung VAL\_STRG

Tabelle 8- 28 Anweisung Zahlenwert in Zeichenkette wandeln

KOP/FUP	SCL	Beschreibung
	<pre>"VAL_STRG" (   in:=_variant_in,   size:=_usint_in,   prec:=_usint_in,   format:=_word_in,   p:=_uint_in,   out=&gt;_string_out);</pre>	<p>Konvertiert einen ganzzahligen Wert mit oder ohne Vorzeichen oder einen Gleitpunktwert in die entsprechende Zeichenkettendarstellung.</p>

<sup>1</sup> In KOP/FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8- 29 Datentypen für die Anweisung VAL\_STRG

Parameter und Datentyp	Datentyp	Beschreibung
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal
SIZE	IN	USInt
PREC	IN	USInt
FORMAT	IN	Word
P	IN	UInt, Byte, USInt
OUT	OUT	String

Der in Parameter IN dargestellte Wert wird in eine Zeichenkette umgewandelt, die in Parameter OUT referenziert wird. Der Parameter OUT muss eine gültige Zeichenkette sein, bevor die Umwandlung erfolgt.



Formatregeln für Zeichenketten in Parameter OUT:

- Führende Leerzeichen werden ganz links an die Zeichenkette angefügt, wenn die umgewandelte Zeichenkette kleiner als die angegebene Größe ist.
- Ist das Vorzeichenbit von Parameter FORMAT FALSCH, so werden ganzzahlige Werte mit und ohne Vorzeichen ohne führendes Vorzeichen "+" in den Ausgangspuffer geschrieben. Vorzeichen "-" wird bei Bedarf verwendet.  
<führende Leerzeichen><Ziffern ohne führende Nullen>'.'  
<PREC-Ziffern>
- Ist das Vorzeichenbit von Parameter WAHR, so werden ganzzahlige Werte mit und ohne Vorzeichen immer mit einem führenden Vorzeichen in den Ausgangspuffer geschrieben.  
<führende Leerzeichen><Vorzeichen><Ziffern ohne führende Nullen>'.'  
<PREC-Ziffern>
- Ist als FORMAT die exponentielle Darstellung eingestellt, so werden Werte des Datentyps Real wie folgt in den Ausgangspuffer geschrieben:  
<führende Leerzeichen><Vorzeichen><Ziffer> '.' <PREC-Ziffern>'E'  
<Vorzeichen><Ziffern ohne führende Null>
- Ist als FORMAT die Festpunktdarstellung eingestellt, so werden ganzzahlige Werte mit und ohne Vorzeichen und Werte des Datentyps Real wie folgt in den Ausgangspuffer geschrieben:  
<führende Leerzeichen><Vorzeichen><Ziffern ohne führende Nullen>'.'  
<PREC-Ziffern>
- Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- Der Wert rechts vom Dezimalpunkt wird gerundet, damit er der mit Parameter PREC eingestellten Anzahl der Ziffern entspricht, die rechts vom Dezimalpunkt angegeben werden sollen.
- Die Größe der ausgegebenen Zeichenkette muss mindestens drei Bytes mehr umfassen als die Anzahl der Ziffern rechts vom Dezimalpunkt.
- Die Werte werden rechtsbündig in der ausgegebenen Zeichenkette angeordnet.

### Mit ENO gemeldete Bedingungen

Tritt während der Umwandlung ein Fehler auf, so werden die folgenden Ergebnisse ausgegeben:

- ENO wird auf 0 gesetzt.
- OUT wird auf 0 gesetzt oder auf den Wert, der in den Beispielen für die Umwandlung von Zeichenketten in Werte angegeben ist.
- OUT bleibt unverändert oder auf dem in den Beispielen gezeigten Wert, wenn OUT eine Zeichenkette ist.

Tabelle 8- 32 ENO-Zustand

ENO	Beschreibung
1	Kein Fehler
0	Unzulässiger oder ungültiger Parameter, zum Beispiel ein Zugriff auf einen DB, der nicht existiert
0	Unzulässige Zeichenkette mit maximaler Länge von 0 oder 255

ENO	Beschreibung
0	Unzulässige Zeichenkette, deren Länge größer als die maximale Länge ist
0	Der umgewandelte Zahlenwert ist zu groß für den für OUT angegebenen Datentyp.
0	Die maximale Zeichenkettengröße für Parameter OUT muss groß genug sein für die Zahl der mit Parameter SIZE angegebenen Zeichen, beginnend mit der Zeichenposition in Parameter P.
0	Unzulässiger Wert P, wobei P=0 oder P größer als die aktuelle Zeichenkettenlänge ist
0	Parameter SIZE muss größer als Parameter PREC sein.

Tabelle 8- 33 Beispiele für die Umwandlung von Zeichenketten in Werte mit S\_CONV

IN-Zeichenkette	Datentyp OUT	OUT-Wert	ENO
"123"	Int oder DInt	123	WAHR
"-00456"	Int oder DInt	-456	WAHR
"123.45"	Int oder DInt	123	WAHR
"+2345"	Int oder DInt	2345	WAHR
"00123AB"	Int oder DInt	123	WAHR
"123"	Real	123.0	WAHR
"123.45"	Real	123.45	WAHR
"1.23e-4"	Real	1.23	WAHR
"1.23E-4"	Real	1.23	WAHR
"12,345.67"	Real	12345.67	WAHR
"3.4e39"	Real	3.4	WAHR
"-3.4e39"	Real	-3.4	WAHR
"1.17549e-38"	Real	1.17549	WAHR
"12345"	SInt	0	FALSCH
"A123"	-/-	0	FALSCH
""	-/-	0	FALSCH
"++123"	-/-	0	FALSCH
"+-123"	-/-	0	FALSCH

Tabelle 8- 34 Beispiele für die Umwandlung von Werten in Zeichenketten mit S\_CONV

Datentyp	IN-Wert	OUT-Zeichenkette	ENO
UInt	123	"123"	WAHR
UInt	0	"0"	WAHR
UDInt	12345678	"12345678"	WAHR
Real	-INF	"INF"	FALSCH
Real	+INF	"INF"	FALSCH
Real	NaN	"NaN"	FALSCH

Tabelle 8- 35 Beispiele für die Umwandlung mit STRG\_VAL

IN-Zeichenkette	FORMAT (W#16#....)	Datentyp OUT	OUT-Wert	ENO
"123"	0000	Int oder DInt	123	WAHR
"-00456"	0000	Int oder DInt	-456	WAHR
"123.45"	0000	Int oder DInt	123	WAHR
"+2345"	0000	Int oder DInt	2345	WAHR
"00123AB"	0000	Int oder DInt	123	WAHR
"123"	0000	Real	123.0	WAHR
"-00456"	0001	Real	-456.0	WAHR
"+00456"	0001	Real	456.0	WAHR
"123.45"	0000	Real	123.45	WAHR
"123.45"	0001	Real	12345.0	WAHR
"123,45"	0000	Real	12345.0	WAHR
"123,45"	0001	Real	123.45	WAHR
".00123AB"	0001	Real	123.0	WAHR
"1.23e-4"	0000	Real	1.23	WAHR
"1.23E-4"	0000	Real	1.23	WAHR
"1.23E-4"	0002	Real	1.23E-4	WAHR
"12,345.67"	0000	Real	12345.67	WAHR
"12,345.67"	0001	Real	12.345	WAHR
"3.4e39"	0002	Real	+INF	WAHR
"-3.4e39"	0002	Real	-INF	WAHR
"1.1754943e-38" (und kleiner)	0002	Real	0.0	WAHR
"12345"	-/-	SInt	0	FALSCH
"A123"	-/-	-/-	0	FALSCH
""	-/-	-/-	0	FALSCH
"++123"	-/-	-/-	0	FALSCH
"+-123"	-/-	-/-	0	FALSCH

Die folgenden Beispiele für VAL\_STRG-Umwandlungen beruhen auf einer Zeichenkette OUT, die wie folgt initialisiert wurde:

```
"Current Temp = xxxxxxxxxx C"
```

Dabei steht das Zeichen "x" für Leerzeichen, die dem umgewandelten Wert zugeordnet sind.

Tabelle 8- 36 Beispiele für die Umwandlung mit VAL\_STRG

Datentyp	IN-Wert	P	SIZE	FORMAT (W#16#....)	PREC	OUT-Zeichenkette	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxx123 C	WAHR
UInt	0	16	10	0000	2	Current Temp = xxxxxx0.00 C	WAHR
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	WAHR
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	WAHR

Datentyp	IN-Wert	P	SIZE	FORMAT (W#16#...)	PREC	OUT-Zeichenkette	ENO
Int	123	16	10	0004	0	Current Temp = xxxxxx+123 C	WAHR
Int	-123	16	10	0004	0	Current Temp = xxxxxx-123 C	WAHR
Real	-0.00123	16	10	0004	4	Current Temp = xxx-0.0012 C	WAHR
Real	-0.00123	16	10	0006	4	Current Temp = -1.2300E-3 C	WAHR
Real	-INF	16	10	-/-	4	Current Temp = xxxxxx-INF C	FALSCH
Real	+INF	16	10	-/-	4	Current Temp = xxxxxx+INF C	FALSCH
Real	NaN	16	10	-/-	4	Current Temp = xxxxxxNaN C	FALSCH
UDInt	12345678	16	6	-/-	3	Current Temp = xxxxxxxxxxxx C	FALSCH

### 8.2.3.2 Umwandlung Zeichenkette-in-Zeichen und Zeichen-in-Zeichenkette

Chars\_TO\_Strg kopiert ein Array aus ASCII-Zeichen-Bytes in eine Zeichenkette.

Strg\_TO\_Chars kopiert eine ASCII-Zeichenkette in ein Array aus Zeichenbytes.

---

#### Hinweis

Nur die nullbasierten Array-Typen (Array [0..n] of Char) oder (Array [0..n] of Byte) sind als Eingangsparameter Chars bei der Anweisung Chars\_TO\_Strg zulässig, oder als IN\_OUT-Parameter Chars bei der Anweisung Strg\_TO\_Chars .

---

Tabelle 8- 37 Anweisung Chars\_TO\_Strg

KOP/FUP	SCL	Beschreibung
	<pre>Chars_TO_Strg(   Chars:=_variant_in_,   pChars:=_dint_in_,   Cnt:=_uint_in_,   Strg=&gt;_string_out_);</pre>	<p>Das gesamte Array oder ein Teil eines Arrays aus Zeichen wird in eine Zeichenkette kopiert.</p> <p>Die Ausgangszeichenkette muss deklariert sein, bevor Chars_TO_Strg ausgeführt wird. Die Zeichenkette wird dann von der Anweisung Chars_TO_Strg überschrieben.</p> <p>Zeichenketten aller unterstützten maximalen Längen (1 ... 254) können verwendet werden.</p> <p>Der Wert für die maximale Länge der Zeichenkette wird von der Anweisung Chars_TO_Strg nicht verändert. Das Kopieren aus dem Array in die Zeichenkette endet, wenn die maximale Länge der Zeichenkette erreicht ist.</p> <p>Das nul-Zeichen "\$00" oder der Wert 16#00 im Zeichen-Array fungiert als Trennzeichen und beendet das Kopieren von Zeichen in die Zeichenkette.</p>

Tabelle 8- 38 Datentypen für die Parameter (Chars\_TO\_Strg)

Parameter und Datentyp		Datentyp	Beschreibung
Chars	IN	Variante	Der Parameter Chars ist ein Pointer auf das nullbasierte Array [0..n] aus Zeichen, das in eine Zeichenkette umgewandelt werden soll. Das Array kann in einem DB oder als lokale Variablen in der Bausteinschnittstelle deklariert werden. Beispiel: "DB1".MyArray verweist auf die Elementwerte MyArray [0..10] of Char in DB1.
pChars	IN	DInt	Elementnummer des ersten zu kopierenden Zeichens im Array. Array-Element [0] ist der Standardwert.
Cnt	IN	UInt	Anzahl der zu kopierenden Zeichen: 0 bedeutet alle
Strg	OUT	String	Zielzeichenkette

Tabelle 8- 39 Anweisung Strg\_TO\_Chars

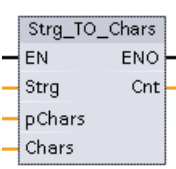
KOP/FUP	SCL	Beschreibung
	<pre>Strg_TO_Chars (   Strg:=_string_in_,   pChars:=_dint_in_,   Cnt=&gt;_uint_out_,   Chars:=_variant_inout_);</pre>	<p>Die vollständige Eingangszeichenkette Strg wird in ein Array aus Zeichen am IN_OUT-Parameter Chars. kopiert.</p> <p>Die Anweisung überschreibt Bytes beginnend mit der Array-Elementnummer, die vom Parameter pChars angegeben wird.</p> <p>Zeichenketten aller unterstützten maximalen Längen (1 .. 254) können verwendet werden.</p> <p>Es wird kein Abschlusstrennzeichen geschrieben, hierfür sind Sie selbst verantwortlich. Wenn Sie das Abschlusstrennzeichen direkt nach dem letzten geschriebenen Array-Zeichen setzen möchten, verwenden Sie die nächste Array-Elementnummer [pChars+Cnt].</p>

Tabelle 8- 40 Datentypen für die Parameter (Strg\_TO\_Chars)

Parameter und Datentyp		Datentyp	Beschreibung
Strg	IN	String	Quellzeichenfolge
pChars	IN	DInt	Array-Elementnummer des ersten Zeichens der Zeichenkette, das in das Ziel-Array geschrieben wird
Chars	IN_OUT	Variante	Der Parameter Chars ist ein Pointer auf das nullbasierte Array [0..n] aus Zeichen, das aus der Eingangszeichenkette kopiert wird. Das Array kann in einem DB oder als lokale Variablen in der Bausteinschnittstelle deklariert werden. Beispiel: "DB1".MyArray verweist auf die Elementwerte MyArray [0..10] of Char in DB1.
Cnt	OUT	UInt	Anzahl der kopierten Zeichen

Tabelle 8- 41 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Chars_TO_Strg: Versuch, mehr Zeichenbytes in die Ausgangszeichenkette zu kopieren als entsprechend dem Byte für die maximale Länge in der Zeichenkettendeklaration zulässig sind
0	Chars_TO_Strg: Der Nullzeichenwert (16#00) wurde im Byte-Array des Eingangszeichens gefunden.
0	Strg_TO_Chars: Versuch, mehr Zeichenbytes in das Ausgangs-Array zu kopieren als entsprechend dem Grenzwert für die Elementanzahl zulässig sind

### 8.2.3.3 Umwandlung von ASCII-Zeichenketten in Hexadezimalzahlen und von Hexadezimalzahlen in ASCII-Zeichenketten

Mit den Anweisungen ATH (ASCII-Zeichenkette in Hexadezimalzahl wandeln) und HTA (Hexadezimalzahl in ASCII-Zeichenkette wandeln) können Sie ASCII-Zeichen (0 bis 9 und Großbuchstaben A bis F) in die entsprechenden 4-Bit-Hexadezimaldarstellungen umwandeln und umgekehrt.

Tabelle 8- 42 Anweisung ATH

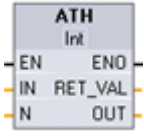
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := ATH(     in:=_variant_in_,     n:=_int_in_,     out=&gt;_variant_out_);</pre>	Konvertiert ASCII-Zeichen in gepackte Hexadezimalziffern.

Tabelle 8- 43 Datentypen für die Anweisung ATH

Parametertyp		Datentyp	Beschreibung
IN	IN	Variante	Pointer zum ASCII-Zeichenbyte-Datenfeld
N	IN	UInt	Anzahl der ASCII-Zeichenbytes, die umzuwandeln sind
RET_VAL	OUT	Word	Ausführungsbedingung
OUT	OUT	Variante	Zeigt zum umgewandelten Hexadezimalbyte-Datenfeld

Die Umwandlung beginnt an der von Parameter IN festgelegten Stelle und umfasst N Bytes. Das Ergebnis wird an der in Parameter OUT angegebenen Adresse ausgegeben. Es können nur gültige ASCII-Zeichen von 0 bis 9 und die Großbuchstaben A bis F umgewandelt werden. Alle anderen Zeichen werden in Null umgewandelt.

8-Bit ASCII-codierte Zeichen werden in 4-Bit Hexadezimalzeichen umgewandelt. Es können zwei ASCII-Zeichen in einem Byte gespeichert werden.

Die Parameter IN und OUT geben Byte-Arrays, keine hexadezimalen String-Daten an. Die Umwandlung der ASCII-Zeichen und ihre Ausgabe in Hexadezimalzeichen erfolgt in der Reihenfolge, in der sie gelesen werden. Bei einer ungeraden Anzahl ASCII-Zeichen werden Nullen in das äußerste rechte Halbbyte der zuletzt umgewandelten Hexadezimalstelle geschrieben.



Tabelle 8- 44 Beispiele für die Umwandlung von ASCII-Zeichenketten in Hexadezimalzahlen (ATH)

IN-Zeichenbytes	N	OUT-Wert	ENO
'0123'	4	W#16#0123	WAHR
'123AFx1a23'	10	16#123AF01023	FALSCH
'a23'	3	W#16#A230	WAHR

Tabelle 8- 45 Anweisung HTA

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := HTA(     in:=_variant_in_,     n:=_uint_in_,     out=&gt;_variant_out_);</pre>	Konvertiert gepackte Hexadezimalziffern in die entsprechenden ASCII-Zeichenbytes.

Tabelle 8- 46 Datentypen für die Anweisung HTA

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
IN	IN	Variante	Zeigt zum Eingangsbyte-Datenfeld
N	IN	UInt	Anzahl der Bytes, die umgewandelt werden sollen (jedes Eingangsbyte hat zwei 4-Bit-Halbbytes und erzeugt 2N ASCII-Zeichen)
RET_VAL	OUT	Word	Ausführungsbedingung
OUT	OUT	Variante	Pointer zum ASCII-Zeichenbyte-Datenfeld

Die Umwandlung beginnt an der von Parameter IN festgelegten Stelle und umfasst N Bytes. Jedes 4-Bit-Halbbyte wird in ein einzelnes 8-Bit-ASCII-Zeichen umgewandelt und erzeugt 2N-ASCII-Zeichenbytes am Ausgang. Alle 2N Bytes des Ausgangs werden als ASCII-Zeichen 0 bis 9 bzw. Großbuchstaben A bis F geschrieben. Der Parameter OUT gibt ein Byte-Array, keine Zeichenkette, an.

Jedes Halbbyte des Hexadezimalbytes wird in ein Zeichen in der Reihenfolge des Einlesens umgewandelt (das linke Halbbyte einer Hexadezimalziffer wird zuerst umgewandelt, dann das rechte Halbbyte des gleichen Bytes).

Tabelle 8- 47 Beispiele für die Umwandlung von Hexadezimalzahlen in ASCII-Zeichenketten (HTA)

IN-Wert	N	OUT-Zeichenbytes	ENO (ENO immer WAHR nach Ausführung von HTA)
W#16#0123	2	'0123'	WAHR
DW#16#123AF012	4	'123AF012'	WAHR

Tabelle 8- 48 ATH and HTA-Bedingungscodes

RET_VAL (W#16#....)	Beschreibung	ENO
0000	Kein Fehler	WAHR
0007	Ungültiges ATH-Eingangszeichen: Es wurde ein Zeichen gefunden, bei dem es sich nicht um ein ASCII-Zeichen 0-9, einen Kleinbuchstaben a-f oder um einen Großbuchstaben A-F handelte	FALSCH
8101	Unzulässiger oder ungültiger Eingangspointer, zum Beispiel ein Zugriff auf einen DB, der nicht existiert.	FALSCH
8120	Eingangszeichenkette hat ein ungültiges Format, d. h. Max. = 0, Max. = 255, aktuell > Max. oder Länge im Pointer < Max.	FALSCH
8182	Eingangspuffer ist zu klein für N	FALSCH
8151	Datentyp für Eingangspuffer nicht zulässig	FALSCH
8301	Unzulässiger oder ungültiger Ausgangspointer, zum Beispiel ein Zugriff auf einen DB, der nicht existiert.	FALSCH
8320	Ausgangszeichenkette hat ein ungültiges Format, d. h. Max. = 0, Max. = 255, aktuell > Max. oder Länge im Pointer < Max.	FALSCH
8382	Ausgangspuffer ist zu klein für N	FALSCH
8351	Datentyp für Ausgangspuffer nicht zulässig	FALSCH

## 8.2.4 Zeichenkettenanweisungen

Ihr Steuerungsprogramm kann mit den folgenden Zeichenketten- und Zeichenanweisungen Meldungen für Bedieneranzeigen und Prozessprotokolle erzeugen.

### 8.2.4.1 LEN

Tabelle 8- 49 Anweisung LEN


KOP/FUP	SCL	Beschreibung
	<pre>out := LEN(in);</pre>	LEN (Länge) gibt die aktuelle Länge der Zeichenkette IN am Ausgang OUT aus. Eine leere Zeichenkette hat die Länge Null.

Tabelle 8- 50 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	String	Eingegebene Zeichenkette
OUT	OUT	Int, DInt, Real, LReal	Anzahl gültiger Zeichen in Zeichenkette IN

Tabelle 8- 51 ENO-Zustand

ENO	Bedingung	OUT
1	Keine gültige Zeichenkette	Gültige Zeichenkettenlänge
0	Aktuelle Länge von IN überschreitet die maximale Länge von IN	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN passt nicht in den zugewiesenen Speicherbereich	
	Maximale Länge von IN ist 255 (unzulässige Länge)	

### 8.2.4.2 CONCAT

Tabelle 8- 52 Anweisung Zeichenketten verketten


KOP/FUP	SCL	Beschreibung
	<pre>out := CONCAT(in1, in2);</pre>	CONCAT (Zeichenketten verketten) verknüpft die String-Parameter IN1 und IN2 zu einer Zeichenkette, die am Ausgang OUT bereitsteht. Nach der Verknüpfung ist Zeichenkette IN1 der linke und Zeichenkette IN2 der rechte Teil der kombinierten Zeichenkette.

Tabelle 8- 53 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
IN1	IN	String	Eingegebene Zeichenkette 1
IN2	IN	String	Eingegebene Zeichenkette 2
OUT	OUT	String	Kombinierte Zeichenkette (Zeichenkette 1 + Zeichenkette 2)

Tabelle 8- 54 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	Resultierende Zeichenkette nach Verknüpfung ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	Die aktuelle Länge von IN1 überschreitet die maximale Länge von IN1, die aktuelle Länge von IN2 überschreitet die maximale Länge von IN2 oder die aktuelle Länge von OUT überschreitet die maximale Länge von OUT (ungültige Zeichenkette)	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
	Maximale Länge von IN1 oder IN2 ist 255 oder die maximale Länge von OUT ist 0 oder 255	

8.2.4.3 LEFT, RIGHT und MID

Tabelle 8- 55 Anweisungen LEFT, RIGHT und MID

KOP/FUP	SCL	Beschreibung
	<pre>out := LEFT(in, L);</pre>	<p>LEFT (Linke Teilzeichenkette) gibt eine Teilzeichenkette aus, die aus den ersten L Zeichen von Parameter IN besteht.</p> <ul style="list-style-type: none"> <li>Ist L größer als die aktuelle Länge der Zeichenkette IN, so wird die gesamte Zeichenkette IN an Ausgang OUT ausgegeben.</li> <li>Wird eine leere Zeichenkette eingegeben, so wird auch eine leere Zeichenkette an OUT ausgegeben.</li> </ul>
	<pre>out := MID(in, L, p);</pre>	<p>MID (Mittlere Teilzeichenkette) gibt den mittleren Teil einer Zeichenkette aus. Die mittlere Teilzeichenkette ist L Zeichen lang und beginnt an Zeichenposition P (einschließlich).</p> <p>Wenn die Summe aus L und P die aktuelle Länge des String-Parameters IN überschreitet, wird eine Teilzeichenkette von Zeichenposition P bis zum Ende von Zeichenkette IN ausgegeben.</p>
	<pre>out := RIGHT(in, L);</pre>	<p>RIGHT (Rechte Teilzeichenkette) gibt die letzten L Zeichen einer Zeichenkette aus.</p> <ul style="list-style-type: none"> <li>Ist L größer als die aktuelle Länge der Zeichenkette IN, so wird die gesamte Zeichenkette IN an Ausgang OUT ausgegeben.</li> <li>Wird eine leere Zeichenkette eingegeben, so wird auch eine leere Zeichenkette an OUT ausgegeben.</li> </ul>

Tabelle 8- 56 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
IN	IN	String	Eingegebene Zeichenkette
L	IN	Int	Länge der zu erstellenden Teilzeichenkette: <ul style="list-style-type: none"> <li>LEFT verwendet die Anzahl der Zeichen ganz links in der Zeichenkette</li> <li>RIGHT verwendet die Anzahl der Zeichen ganz rechts in der Zeichenkette</li> <li>MID verwendet die Anzahl der Zeichen ab Position P in der Zeichenkette</li> </ul>
P	IN	Int	Nur MID: Position des ersten Zeichens in der Teilzeichenkette, das kopiert werden soll P = 1, für die Position des ersten Zeichens der Zeichenkette IN
OUT	OUT	String	Ausgegebene Zeichenkette

Tabelle 8- 57 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	<ul style="list-style-type: none"> <li>L oder P ist kleiner oder gleich 0</li> <li>P ist größer als die maximale Länge von IN</li> <li>Aktuelle Länge von IN überschreitet maximale Länge von IN, oder aktuelle Länge von OUT überschreitet maximale Länge von OUT</li> <li>Maximale Länge von IN oder OUT passt nicht in den zugewiesenen Speicher</li> <li>Maximale Länge von IN oder OUT ist 0 oder 255</li> </ul>	Aktuelle Länge wird auf 0 gesetzt
	Länge (L) der Teilzeichenkette, die kopiert werden soll, ist größer als die maximale Länge von Zeichenkette OUT	Es werden so viele Zeichen kopiert, bis die maximale Länge von OUT erreicht ist
	Nur MID: L oder P ist kleiner oder gleich 0	Aktuelle Länge wird auf 0 gesetzt
	Nur MID: P ist größer als die maximale Länge von IN	
	Aktuelle Länge von IN1 überschreitet maximale Länge von IN1 oder aktuelle Länge von IN2 überschreitet maximale Länge von IN2 (ungültige Zeichenkette)	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
	Maximale Länge von IN1, IN2 oder OUT ist 0 oder 255 (unzulässige Länge)	

### 8.2.4.4 DELETE

Tabelle 8- 58 Anweisung Teilzeichenkette löschen

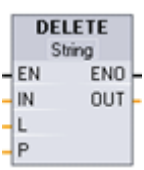
KOP/FUP	SCL	Beschreibung
	<pre>out := DELETE(in, L, p);</pre>	<p>Löscht L Zeichen aus der Zeichenkette IN. Mit dem Löschen der Zeichen wird an Zeichenposition P (einschließlich) begonnen, und die restliche Teilzeichenkette wird an Parameter OUT ausgegeben.</p> <ul style="list-style-type: none"> <li>Ist L gleich Null, so wird die eingegebene Zeichenkette an OUT ausgegeben.</li> <li>Ist die Summe aus L und P größer als die Länge der eingegebenen Zeichenkette, so wird die Zeichenkette bis zum Ende gelöscht.</li> </ul>

Tabelle 8- 59 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
IN	IN	String
L	IN	Int

Parameter und Datentyp		Datentyp	Beschreibung
P	IN	Int	Position des ersten Zeichens, das gelöscht werden soll: Das erste Zeichen der Zeichenkette IN hat Position 1.
OUT	OUT	String	Ausgegebene Zeichenkette

Tabelle 8- 60 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	P ist größer als die aktuelle Länge von IN	IN wird in OUT kopiert, es werden keine Zeichen gelöscht
	Resultierende Zeichenkette nach Löschen ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	L ist kleiner als 0 oder P ist kleiner oder gleich 0	Aktuelle Länge wird auf 0 gesetzt
	Aktuelle Länge von IN überschreitet maximale Länge von IN, oder aktuelle Länge von OUT überschreitet maximale Länge von OUT	
	Maximale Länge von IN oder OUT passt nicht in den zugewiesenen Speicher	
	Maximale Länge von IN oder OUT ist 0 oder 255	

### 8.2.4.5 INSERT

Tabelle 8- 61 Anweisung Teilzeichenkette einfügen

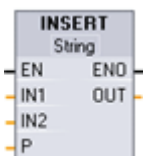
KOP/FUP	SCL	Beschreibung
	<pre>out := INSERT(in1, in2, p);</pre>	Fügt die Zeichenkette IN2 in die Zeichenkette IN1 ein. Das Einfügen beginnt nach dem Zeichen an Position P.

Tabelle 8- 62 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
IN1	IN	String	Eingegebene Zeichenkette 1
IN2	IN	String	Eingegebene Zeichenkette 2
P	IN	Int	Letzte Zeichenposition in Zeichenkette IN1 vor dem Einfügungspunkt für Zeichenkette IN2. Das erste Zeichen der Zeichenkette IN1 hat Position 1.
OUT	OUT	String	Resultierende Zeichenkette

Tabelle 8- 63 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	P ist größer als die Länge von IN1	IN2 wird unmittelbar nach dem letzten Zeichen von IN1 mit IN1 verkettet
	P ist kleiner als 0	Aktuelle Länge wird auf 0 gesetzt
	Resultierende Zeichenkette nach Einfügen ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	Die aktuelle Länge von IN1 überschreitet die maximale Länge von IN1, die aktuelle Länge von IN2 überschreitet die maximale Länge von IN2 oder die aktuelle Länge von OUT überschreitet die maximale Länge von OUT (ungültige Zeichenkette)	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
	Maximale Länge von IN1 oder IN2 ist 255 oder die maximale Länge von OUT ist 0 oder 255	

### 8.2.4.6 REPLACE

Tabelle 8- 64 Anweisung Teilzeichenkette ersetzen

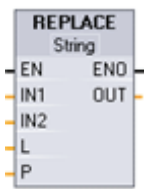
KOP/FUP	SCL	Beschreibung
	<pre> out := REPLACE (     in1:=_string_in_,     in2:=_string_in_,     L:=_int_in_,     p:=_int_in); </pre>	Ersetzt L Zeichen in der Zeichenkette im Parameter IN1. Das Ersetzen beginnt an Zeichenposition P (einschließlich) der Zeichenkette IN1, die Ersetzungszeichen werden aus Parameter IN2 übernommen.

Tabelle 8- 65 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
IN1	IN	String	Eingegebene Zeichenkette
IN2	IN	String	Zeichenkette mit Ersatzzeichen
L	IN	Int	Anzahl der Zeichen, die ersetzt werden sollen
P	IN	Int	Position des ersten Zeichens, das ersetzt werden soll
OUT	OUT	String	Resultierende Zeichenkette

Ist Parameter L gleich Null, so wird Zeichenkette IN2 an Position P von Zeichenkette IN1 eingefügt, ohne dass Zeichen in Zeichenkette IN1 gelöscht werden.

Ist P gleich 1, werden die ersten L Zeichen von Zeichenkette IN1 durch die Zeichen der Zeichenkette IN2 ersetzt.

Tabelle 8- 66 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	P ist größer als die Länge von IN1	IN2 wird unmittelbar nach dem letzten Zeichen von IN1 mit IN1 verkettet
	P Punkte in IN1, aber weniger als L Zeichen in IN1	IN2 ersetzt das Endezeichen von IN1 ab Position P
	Resultierende Zeichenkette nach Ersetzen ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	Maximale Länge von IN1 ist 0	IN2 Zeichen werden in OUT kopiert
	L ist kleiner als 0 oder P ist kleiner oder gleich 0	Aktuelle Länge wird auf 0 gesetzt
	Die aktuelle Länge von IN1 überschreitet die maximale Länge von IN1, die aktuelle Länge von IN2 überschreitet die maximale Länge von IN2 oder die aktuelle Länge von OUT überschreitet die maximale Länge von OUT	
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
	Maximale Länge von IN1 oder IN2 ist 255 oder die maximale Länge von OUT ist 0 oder 255	

8.2.4.7 FIND

Tabelle 8- 67 Anweisung Teilzeichenkette finden

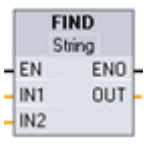
KOP/FUP	SCL	Beschreibung
	<pre> out := FIND(     in1:=_string_in_,     in2:=_string_in_);         </pre>	Liefert die Zeichenposition der von IN2 angegebenen Zeichenkette in der Zeichenkette IN1. Die Suche beginnt links. Die Zeichenposition des ersten Vorkommens von IN2 wird an OUT ausgegeben. Wird die Zeichenkette IN2 nicht in der Zeichenkette IN1 gefunden, so wird Null ausgegeben.

Tabelle 8- 68 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
IN1	IN	String	In dieser Zeichenkette suchen
IN2	IN	String	Diese Zeichenkette suchen
OUT	OUT	Int	Zeichenposition in Zeichenkette IN1, erstes positives Suchergebnis



Tabelle 8- 69 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichenposition
0	IN2 ist größer als IN1	Zeichenposition wird auf 0 gesetzt
	Aktuelle Länge von IN1 überschreitet maximale Länge von IN1 oder aktuelle Länge von IN2 überschreitet maximale Länge von IN2 (ungültige Zeichenkette)	
	Maximale Länge von IN1 oder IN2 passt nicht in den zugewiesenen Speicherbereich	
	Maximale Länge von IN1 oder IN2 ist 255	

## 8.3 Dezentrale E/A (PROFINET, PROFIBUS oder AS-i)

### 8.3.1 Anweisungen für die dezentrale E/A

Die folgenden Anweisungen für die dezentrale Peripherie können entweder mit PROFINET, PROFIBUS oder AS-i verwendet werden:

- Anweisung RDREC (Seite 290): Mit der Nummer INDEX aus einem Modul oder Gerät können Sie einen Datensatz lesen.
- Anweisung WRREC (Seite 290): Sie können einen Datensatz mit der Nummer INDEX in ein durch ID angegebenes Modul oder Gerät übertragen.
- Anweisung RALRM (Seite 293): Sie können einen Alarm mit allen entsprechenden Informationen von einem Modul oder Gerät empfangen und diese Informationen an die Ausgangsparameter übergeben.
- Anweisung DPRD\_DAT (Seite 300): Konsistente Datenbereiche größer als 64 Byte müssen Sie über die Anweisung DPRD\_DAT aus einem Modul oder Gerät auslesen.
- Anweisung DPWR\_DAT (Seite 300): Konsistente Datenbereiche größer als 64 Byte müssen Sie über die Anweisung DPWR\_DAT in ein Modul oder Gerät schreiben.

Die Anweisung DPNRM\_DG (Seite 302) kann nur mit PROFIBUS verwendet werden. Die aktuellen Diagnosedaten eines DP-Slaves können Sie in dem Format auslesen, das in EN 50 170 Band 2, PROFIBUS, angegeben ist.

### 8.3.2 RDREC und WRREC

Die Anweisungen RDREC (Datensatz lesen) und WRREC (Datensatz schreiben) können Sie für PROFINET, PROFIBUS und AS-i einsetzen.

Tabelle 8- 70 Anweisungen RDREC und WRREC

KOP/FUP	SCL	Beschreibung
<p>"RDREC_DB"</p>	<pre>"RDREC_DB" (   req:=_bool_in_,   ID:=_word_in_,   index:=_dint_in_,   mlen:=_uint_in_,   valid=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   len=&gt;_uint_out_,   record:=_variant_inout_);</pre>	<p>Mit der Anweisung RDREC lesen Sie einen Datensatz mit der Nummer INDEX aus der Komponente, die von ID angegeben wird, z. B. aus einem zentralen Baugruppenträger oder einer dezentralen Komponente (PROFIBUS DP oder PROFINET IO). Geben Sie die maximale Anzahl der zu lesenden Bytes in MLEN an. Die ausgewählte Länge des Zielbereichs RECORD muss mindestens die Länge der von MLEN angegebenen Bytes haben.</p>
<p>"WRREC_DB"</p>	<pre>"WRREC_DB" (   req:=_bool_in_,   ID:=_word_in_,   index:=_dint_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   record:=_variant_inout_);</pre>	<p>Mit der Anweisung WRREC übertragen Sie einen Datensatz (RECORD) mit der Nummer INDEX an einen DP-Slave bzw. eine PROFINET IO-Gerätekompone, der/die von ID angegeben wird, z. B. ein Modul im zentralen Baugruppenträger oder an eine dezentrale Komponente (PROFIBUS DP oder PROFINET IO). Geben Sie die Byte-Länge des zu übertragenden Datensatzes an. Die ausgewählte Länge des Quellbereichs RECORD muss deshalb mindestens die Länge der von LEN angegebenen Bytes haben.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 In den SCL-Beispielen sind "RDREC\_DB" und "WRREC\_DB" die Namen der Instanz-DBs.

Tabelle 8- 71 Datentypen für die Parameter von RDREC und WRREC

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	REQ = 1: Datensatz übertragen
ID	IN	HW_IO (Word)	<p>Logische Adresse der DP-Slave/PROFINET IO-Komponente (Modul oder Submodul):</p> <ul style="list-style-type: none"> <li>• Bei einem Ausgabemodul muss Bit 15 gesetzt werden (zum Beispiel für Adresse 5: ID=DW#16#8005).</li> <li>• Bei einem Kombimodul muss die kleinere der beiden Adressen angegeben werden.</li> </ul> <p><b>Hinweis:</b> Die Geräte-ID kann wahlweise auf zwei Arten ermittelt werden:</p> <ul style="list-style-type: none"> <li>• Durch Auswahl der folgenden Optionen in der "Netzschritt": <ul style="list-style-type: none"> <li>– Gerät (graues Feld)</li> <li>– "Eigenschaften" des Geräts</li> <li>– "Hardwarekennung" <p><b>Hinweis:</b> Es wird jedoch nicht für alle Geräte die Hardwarekennung angezeigt.</p> </li> </ul> </li> <li>• Durch Auswahl der folgenden Optionen in der "Projektnavigation": <ul style="list-style-type: none"> <li>– PLC-Variablen</li> <li>– Standardvariablen-Tabelle</li> <li>– Register "Konstanten" des Systems</li> </ul> </li> </ul> <p>Die Hardwarekennungen aller konfigurierten Geräte werden angezeigt.</p>
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	Datensatznummer
MLEN	IN	Byte, USInt, UInt	Maximale Länge in Bytes der abzurufenden Informationen aus dem Datensatz (RDREC)
VALID	OUT	Bool	Neuer Datensatz wurde empfangen und ist gültig (RDREC). Das Bit VALID ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
DONE	OUT	Bool	Datensatz wurde übertragen (WRREC). Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• BUSY = 1: Der Lesevorgang (RDREC) oder Schreibvorgang (WRREC) ist noch nicht beendet.</li> <li>• BUSY = 0: Die Übertragung des Datensatzes ist beendet.</li> </ul>
ERROR	OUT	Bool	ERROR = 1: Ein Lesefehler (RDREC) bzw. Schreibfehler (WRREC) ist aufgetreten. Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	DWord	Bausteinstatus oder Fehlerinformationen

Parameter und Datentyp		Datentyp	Beschreibung
LEN	OUT (RDREC) IN (WRREC)	UInt	<ul style="list-style-type: none"> <li>• Länge der abgerufenen Informationen aus dem Datensatz (RDREC)</li> <li>• Maximale Byte-Länge des zu übertragenden Datensatzes (WRREC)</li> </ul>
RECORD	IN_OUT	Variant	<ul style="list-style-type: none"> <li>• Zielbereich für den abgerufenen Datensatz (RDREC)</li> <li>• Datensatz (WRREC)</li> </ul>

Die Anweisungen RDREC und WRREC funktionieren asynchron, d.h. die Verarbeitung deckt mehrere Anweisungsaufrufe ab. Sie starten den Auftrag durch Aufruf von RDREC oder WRREC mit REQ = 1.

Der Auftragsstatus wird über den Ausgangsparameter BUSY und die beiden zentralen Bytes des Ausgangsparameters STATUS angezeigt. Die Übertragung des Datensatzes ist beendet, wenn der Ausgangsparameter BUSY den Wert FALSE hat.

Ist der Ausgangsparameter VALID (RDREC) oder DONE (WRREC) gleich TRUE (nur einen Zyklus lang), weist dies darauf hin, dass der Datensatz erfolgreich in den Zielbereich RECORD (RDREC) oder zum Zielgerät (WRREC) übertragen wurde. Bei RDREC enthält der Ausgangsparameter LEN die Länge der abgerufenen Daten in Bytes.

Der Ausgangsparameter ERROR (nur einen Zyklus lang, wenn ERROR = TRUE) zeigt an, dass beim Übertragen des Datensatzes ein Fehler aufgetreten ist. In diesem Fall enthält der Ausgangsparameter STATUS (nur während des einen Zyklus, in dem ERROR = TRUE) die Fehlerinformationen.

Datensätze werden vom Hersteller des Hardwaregeräts definiert. Ausführliche Informationen zu einem Datensatz finden Sie in der Gerätedokumentation des jeweiligen Hardwareherstellers.

---

**Hinweis**

Wenn ein DPV1-Slave über eine GSD-Datei (GSD ab Version 3) konfiguriert ist und die DP-Schnittstelle des DP-Masters auf "S7-kompatibel" gesetzt ist, dann können Sie im Anwenderprogramm mit "RDREC" keine Datensätze aus den E/A-Modulen lesen und mit "WRREC" keine Datensätze in die E/A-Module schreiben. In einem solchen Fall adressiert der DP-Master den falschen Steckplatz (konfigurierter Steckplatz + 3).

Abhilfe: Richten Sie als Schnittstelle des DP-Masters "DPV1" ein.

---

**Hinweis**

Die Schnittstellen der Anweisungen "RDREC" und "WRREC" sind mit den FBs "RDREC" und "WRREC" identisch, die in der "PROFIBUS-Richtlinie für die PROFIBUS-Kommunikation und Proxy-Funktionsbausteine nach IEC 61131-3" definiert sind.

---

**Hinweis**

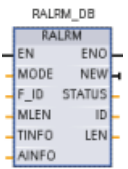
Wenn Sie "RDREC" oder "WRREC" verwenden, um einen Datensatz für PROFINET IO zu lesen oder zu schreiben, werden negative Werte in den Parametern INDEX, MLEN und LEN als vorzeichenlose 16-Bit-Ganzzahlen interpretiert.

---

### 8.3.3 RALRM

Die Anweisung RALRM (Alarm lesen) können Sie für PROFINET und PROFIBUS einsetzen.

Tabelle 8- 72 Anweisung RALRM

KOP/FUP	SCL	Beschreibung
	<pre>"RALRM_DB" (     mode:=_int_in_,     f_ID:=_word_in_,     mlen:=_uint_in_,     new=&gt;_bool_out_,     status=&gt;_dword_out_,     ID=&gt;_word_out_,     len=&gt;_uint_out_,     tinfo:=_variant_inout_,     ainfo:=_variant_inout_);</pre>	<p>Mit der Anweisung RALRM (Alarm lesen) lesen Sie Diagnosealarminformationen aus einem PROFIBUS- oder PROFINET I/O-Modul/Gerät.</p> <p>Die Informationen in den Ausgangsparametern enthalten die Startinformationen des aufgerufenen OBs sowie die Informationen der Alarmquelle.</p> <p>Rufen Sie RALRM in einem Alarm-OB auf, um zu den Ereignissen, die den Alarm verursacht haben, Informationen auszugeben. Bei der S7-1200 werden nur Diagnosealarme (OB82) unterstützt.</p>

- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "RALRM\_DB" der Name des Instanz-DBs.

Tabelle 8- 73 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
MODE	IN	Byte, USInt, SInt, Int Betriebszustand
F_ID	IN	HW_IO (Word) Logische Anfangsadresse der Komponente (des Moduls), von der Alarme empfangen werden sollen. <b>Hinweis:</b> Die Geräte-ID kann wahlweise auf zwei Arten ermittelt werden: <ul style="list-style-type: none"> <li>Durch Auswahl der folgenden Optionen in der "Netzschritt": <ul style="list-style-type: none"> <li>Gerät (graues Feld)</li> <li>"Eigenschaften" des Geräts</li> <li>"Hardwarekennung" <b>Hinweis:</b> Es wird nicht für alle Geräte die Hardwarekennung angezeigt.</li> </ul> </li> <li>Durch Auswahl der folgenden Optionen in der "Projektnavigation": <ul style="list-style-type: none"> <li>PLC-Variablen</li> <li>Standardvariablen-Tabelle</li> <li>Register "Konstanten" des Systems</li> <li>Die Hardwarekennungen aller konfigurierten Geräte werden angezeigt.</li> </ul> </li> </ul>
MLEN	IN	Byte, USInt, UInt Maximale Länge in Byte der zu empfangenden Informationen aus dem Datensatz. Ist MLEN gleich 0, können so viele Alarminformationen empfangen werden, wie im Zielbereich AINFO zur Verfügung stehen.
NEW	OUT	Bool Ein neuer Alarm wurde empfangen.
STATUS	OUT	DWord Status der Anweisung RALRM. Weitere Informationen finden Sie unter "STATUS-Parameter für RDREC, WRREC und RALRM" (Seite 296).

Parameter und Datentyp		Datentyp	Beschreibung
ID	OUT	HW_IO (Word)	Hardwarekennung des E/A-Moduls, das den Diagnosealarm verursacht hat <b>Hinweis:</b> In der Beschreibung des Parameters F_ID finden Sie eine Erläuterung dazu, wie Sie die Geräte-ID ermitteln.
LEN	OUT	DWord, UInt, UInt, DInt, Real, LReal	Länge der empfangenen Alarminformationen AINFO
TINFO	IN_OUT	Variant	Aufgabeninformationen: Zielbereich für Start- und Verwaltungsinformationen des OBs. Die Länge TINFO ist immer 32 Byte.
AINFO	IN_OUT	Variant	Alarminformationen: Zielbereich für Header-Informationen und zusätzliche Alarminformationen. Für AINFO müssen Sie eine Länge von mindestens der in MLEN angegebenen Bytezahl angeben, wenn MLEN größer als 0 ist. Die Länge AINFO ist variabel.

---

**Hinweis**

Wenn Sie "RALRM" in einem OB aufrufen, dessen Startereignis kein E/A-Alarm ist, liefert die Anweisung entsprechend weniger Informationen in ihren Ausgängen.

Stellen Sie sicher, dass Sie unterschiedliche Instanz-DBs verwenden, wenn Sie "RALRM" in unterschiedlichen OBs aufrufen. Wenn Sie die von einem "RALRM"-Aufruf resultierenden Daten außerhalb des zugehörigen Alarm-OBs auswerten, müssen Sie für jedes OB-Startereignis einen eigenen Instanz-DB verwenden.

---

**Hinweis**

Die Schnittstelle der Anweisung "RALRM" ist mit dem FB "RALRM" identisch, der in der "PROFIBUS-Richtlinie für die PROFIBUS-Kommunikation und Proxy-Funktionsbausteine nach IEC 61131-3" definiert ist.

---

**RALRM aufrufen**

Die Anweisung RALRM können Sie in drei verschiedenen Betriebsarten (MODE) aufrufen.

Tabelle 8- 74 Betriebsarten der Anweisung RALRM

MODE	Beschreibung
0	<ul style="list-style-type: none"> <li>• ID enthält die Hardwarekennung des E/A-Moduls, das den Alarm ausgelöst hat.</li> <li>• Der Ausgangsparameter NEW wird auf WAHR gesetzt.</li> <li>• LEN erzeugt einen Ausgang von 0.</li> <li>• AINFO und TINFO werden nicht mit Informationen aktualisiert.</li> </ul>
1	<ul style="list-style-type: none"> <li>• ID enthält die Hardwarekennung des E/A-Moduls, das den Alarm ausgelöst hat.</li> <li>• Der Ausgangsparameter NEW wird auf WAHR gesetzt.</li> <li>• LEN erzeugt einen Ausgang in der Größe der Byteanzahl der ausgegebenen AINFO-Daten.</li> <li>• AINFO und TINFO werden mit alarmbezogenen Informationen aktualisiert.</li> </ul>
2	<p>Wenn die Hardwarekennung des Eingangsparameters F_ID den Alarm ausgelöst hat, dann gilt Folgendes:</p> <ul style="list-style-type: none"> <li>• ID enthält die Hardwarekennung des E/A-Moduls, das den Alarm ausgelöst hat. Dieser Wert muss mit dem Wert von F_ID übereinstimmen.</li> <li>• Der Ausgangsparameter NEW wird auf WAHR gesetzt.</li> <li>• LEN erzeugt einen Ausgang in der Größe der byteanzahl der ausgegebenen AINFO-Daten.</li> <li>• AINFO und TINFO werden mit alarmbezogenen Informationen aktualisiert.</li> </ul>

**Hinweis**

Wenn Sie einen zu kurzen Zielbereich für TINFO oder AINFO zuweisen, kann RALRM nicht die vollständigen Informationen zurückgeben.

MLEN kann die ausgegebene AINFO-Datenmenge begrenzen.

Informationen dazu, wie Sie die Daten von TINFO und AINFO interpretieren, finden Sie bei den Parametern von AINFO und TINFO im Online-Informationssystem von STEP 7.

### 8.3.4 STATUS-Parameter für RDREC, WRREC und RALRM

Der Ausgangsparameter STATUS enthält Fehlerinformationen, die als ARRAY[1...4] OF BYTE mit der folgenden Struktur ausgewertet werden:

Tabelle 8- 75 Ausgangs-Array STATUS

Array-Element	Name	Beschreibung
STATUS[1]	Function_Num	<ul style="list-style-type: none"> <li>B#16#00, wenn kein Fehler vorliegt</li> <li>Funktions-ID von DPV1-PDU: Tritt ein Fehler auf, wird , B#16#80 durch ODER verknüpft (beim Lesen des Datensatzes mit: B#16#DE; beim Schreiben des Datensatzes mit: B#16#DF). Wenn kein DPV1-Protokollelement verwendet wird, wird B#16#C0 ausgegeben.</li> </ul>
STATUS[2]	Error Decode	Adresse der Fehler-ID
STATUS[3]	Error_Code_1	Fehler-ID
STATUS[4]	Error_Code_2	Herstellerspezifische Erweiterung der Fehler-ID

Tabelle 8- 76 STATUS[2]-Werte

Error_decode (B#16#...)	Quelle	Beschreibung
00 bis 7F	CPU	Kein Fehler und keine Warnung
80	DPV1	Fehler gemäß IEC 61158-6
81 bis 8F	CPU	B#16#8x zeigt einen Fehler im "x-ten" Aufrufparameter der Anweisung.
FE, FF	DP-Profil	Profilspezifischer Fehler

Tabelle 8- 77 STATUS[3]-Werte

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Erläuterung (DVP1)	Beschreibung
00	00		Kein Fehler, keine Warnung
70	00	Reserviert, ablehnen	Erster Aufruf; keine aktive Datensatzübertragung
	01	Reserviert, ablehnen	Erster Aufruf; Datensatzübertragung hat begonnen
	02	Reserviert, ablehnen	Zwischenzeitlicher Aufruf; Datensatzübertragung bereits aktiv
80	90	Reserviert, übergeben	Ungültige logische Anfangsadresse
	92	Reserviert, übergeben	Unzulässiger Typ für Pointer Variant
	93	Reserviert, übergeben	Die über ID oder F_ID adressierte DP-Komponente ist nicht konfiguriert.



Error_decode (B#16#....)	Error_code_1 (B#16#....)	Erläuterung (DVP1)	Beschreibung
	96		Die Anweisung RALRM (Seite 293) kann die Anlaufinformationen, Verwaltungsinformationen, Headerinformationen und weitere Alarminformationen des OBs nicht liefern. Bei den OBs 4x, 55, 56, 57, 82 und 83 können Sie die Anweisung DPNRM_DG (Seite 302) verwenden, um die aktuellen Diagnosemeldungen des betreffenden DP-Slaves asynchron zu lesen (Adressinformationen aus den Anlaufinformationen des OBs).
	A0	Lesefehler	Negative Quittierung beim Lesen aus Modul
	A1	Schreibfehler	Negative Quittierung beim Schreiben in Modul
	A2	Modulausfall	DP-Protokollfehler in Schicht 2 (Beispiel: Slave-Ausfall oder Busprobleme)
	A3	Reserviert, übergeben	<ul style="list-style-type: none"> <li>PROFIBUS DP: DP-Protokollfehler mit Direct-Data-Link-Mapper oder User-Interface/User</li> <li>PROFINET IO: Allgemeiner CM-Fehler</li> </ul>
	A4	Reserviert, übergeben	Kommunikation auf dem Kommunikationsbus unterbrochen
	A5	Reserviert, übergeben	-
	A7	Reserviert, übergeben	DP-Slave oder Module belegt (temporärer Fehler).
	A8	Versionskonflikt	DP-Slave oder Modul meldet nicht kompatible Versionen.
	A9	Funktion nicht unterstützt	Funktion wird von DP-Slave oder Modul nicht unterstützt
	AA bis AF	Benutzerspezifisch	DP-Slave oder Modul meldet herstellerspezifischen Fehler in der Anwendung. Prüfen Sie die Dokumentation des Herstellers des DP-Slaves oder des Moduls.
	B0	Ungültiger Index	Datensatz im Modul unbekannt; unzulässige Datensatznummer $\geq 256$
	B1	Längenfehler beim Schreiben	Die Längeninformationen im Parameter RECORD sind falsch. <ul style="list-style-type: none"> <li>Bei RALRM: Längenfehler in AINFO <b>Hinweis:</b> Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Informationen dazu, wie die von AINFO zurückgegebenen Puffer auszuwerten sind.</li> <li>Bei RDREC (Seite 290) und WRREC (Seite 290): Längenfehler in MLEN</li> </ul>
	B2	Ungültiger Steckplatz	Der konfigurierte Steckplatz ist nicht belegt.
	B3	Typkonflikt	Tatsächlicher Modultyp entspricht nicht dem angegebenen Modultyp.
	B4	Ungültiger Bereich	DP-Slave oder Modul meldet Zugriff auf ungültigen Datenbereich.
	B5	Statuskonflikt	DP-Slave oder Modul nicht betriebsbereit
	B6	Zugriff verweigert	DP-Slave oder Modul verweigert den Zugriff.

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Erläuterung (DVP1)	Beschreibung
	B7	Ungültiger Bereich	DP-Slave oder Modul meldet einen ungültigen Bereich für einen Parameter oder Wert.
	B8	Ungültiger Parameter	DP-Slave oder Modul meldet einen ungültigen Parameter.
	B9	Ungültiger Typ	DP-Slave oder Modul meldet einen ungültigen Typ: <ul style="list-style-type: none"> <li>Bei RDREC (Seite 290): Puffer zu klein (Teilmengen können nicht gelesen werden)</li> <li>Bei WRREC (Seite 290): Puffer zu klein (Teilmengen können nicht geschrieben werden)</li> </ul>
	BA bis BF	Benutzerspezifisch	DP-Slave oder Modul meldet herstellerspezifischen Fehler beim Zugriff. Prüfen Sie die Dokumentation des Herstellers des DP-Slaves oder des Moduls.
	C0	Einschränkungskonflikt beim Lesen	<ul style="list-style-type: none"> <li>Bei WRREC (Seite 290): Die Daten können nur geschrieben werden, wenn sich die CPU in der Betriebsart STOP befindet. <b>Hinweis:</b> Das bedeutet, dass die Daten nicht vom Anwenderprogramm geschrieben werden können. Sie können die Daten nur online mit einem PG/PC schreiben.</li> <li>Bei RDREC (Seite 290): Das Modul leitet den Datensatz, doch es liegen entweder keine Daten vor oder die Daten können nur gelesen werden, wenn sich die CPU in der Betriebsart STOP befindet. <b>Hinweis:</b> Wenn nur in der Betriebsart STOP Daten geschrieben werden können, ist keine Auswertung durch das Anwenderprogramm möglich. Sie können die Daten dann nur online mit einem PG/PC lesen.</li> </ul>
	C1	Einschränkungskonflikt beim Schreiben	Die Daten der vorherigen Schreibanforderung an das Modul für denselben Datensatz wurden vom Modul noch nicht verarbeitet.
	C2	Ressource ausgelastet	Das Modul verarbeitet derzeit die maximal mögliche Anzahl von Aufträgen für eine CPU.
	C3	Ressource nicht verfügbar	Die erforderlichen Betriebsressourcen sind derzeit ausgelastet.
	C4		Interner temporärer Fehler. Der Auftrag konnte nicht durchgeführt werden. Wiederholen Sie den Auftrag. Tritt dieser Fehler häufig auf, prüfen Sie Ihre Installation auf Quellen elektrischer Störung.
	C5		DP-Slave oder Modul nicht verfügbar
	C6		Datensatzübertragung wurde wegen Prioritätsklasseneinordnung abgebrochen.
	C7		Der Auftrag wurde wegen Warmstarts oder Kaltstarts des DP-Masters abgebrochen.
	C8 bis CF		DP-Slave oder Modul meldet herstellerspezifischen Ressourcenfehler. Prüfen Sie die Dokumentation des Herstellers des DP-Slaves oder des Moduls.

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Erläuterung (DVP1)	Beschreibung
	Dx	Benutzerspezifisch	DP-Slave-spezifisch. Siehe Beschreibung des DP-Slave.
81	00 bis FF		Fehler im ersten Aufrufparameter (bei RALRM (Seite 293): MODE)
	00		Unzulässige Betriebsart
82	00 bis FF		Fehler im zweiten Aufrufparameter
88	00 bis FF		Fehler im achten Aufrufparameter (bei RALRM (Seite 293): TINFO) <b>Hinweis:</b> Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Informationen dazu, wie die von TINFO zurückgegebenen Puffer auszuwerten sind.
	01		Falsche Syntax-ID
	23		Quantitätsstruktur überschritten oder Zielbereich zu klein
	24		Falsche Bereichs-ID
	32		DB/DI-Nummer außerhalb des Anwenderbereichs
	3A		DB/DI-Nummer ist NULL für die Bereichs-ID DB/DI oder der angegebene DB/DI existiert nicht.
89	00 bis FF		Fehler im neunten Aufrufparameter (bei RALRM (Seite 293): AINFO) <b>Hinweis:</b> Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Informationen dazu, wie die von AINFO zurückgegebenen Puffer auszuwerten sind.
	01		Falsche Syntax-ID
	23		Quantitätsstruktur überschritten oder Zielbereich zu klein
	24		Falsche Bereichs-ID
	32		DB/DI-Nummer außerhalb des Anwenderbereichs
	3A		DB/DI-Nummer ist NULL für die Bereichs-ID DB/DI oder der angegebene DB/DI existiert nicht.
8A	00 bis FF		Fehler im zehnten Aufrufparameter
8F	00 bis FF		Fehler im fünfzehnten Aufrufparameter
FE, FF	00 bis FF		Profilspezifischer Fehler

#### Array-Element STATUS[4]

Bei DPV1-Fehlern übergibt der DP-Master STATUS[4] an die CPU und die Anweisung. Ohne DPV1-Fehler ist dieser Wert auf 0 gesetzt, mit den folgenden Ausnahmen bei RDREC:

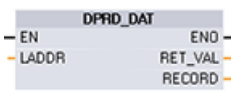

- STATUS[4] enthält die Zielbereichslänge von RECORD, wenn MLEN > die Zielbereichslänge aus RECORD.
- STATUS[4] = MLEN, wenn die tatsächliche Länge des Datensatzes < MLEN < die Länge des Zielbereichs aus RECORD.
- STATUS[4] = 0, wenn STATUS[4] > 255; müsste gesetzt werden

Bei PROFINET IO hat STATUS[4] den Wert 0.

### 8.3.5 DPRD\_DAT und DPWR\_DAT

Die Anweisungen DPRD\_DAT (Konsistente Daten lesen) und DPWR\_DAT (Konsistente Daten schreiben) können Sie für PROFINET und PROFIBUS einsetzen.

Tabelle 8- 78 Anweisungen DPRD\_DAT und DPWR\_DAT

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := DPRD_DAT(     laddr:=_word_in_,     record=&gt;_variant_out_);</pre>	<p>Mit der Anweisung DPRD_DAT können Sie die konsistenten Daten eines DP-Normslaves bzw. PROFINET IO-Geräts lesen. Tritt während der Datenübertragung kein Fehler auf, werden die gelesenen Daten in den vom Parameter RECORD angegebenen Zielbereich eingegeben. Der Zielbereich muss dieselbe Länge haben wie die, die Sie für das ausgewählte Modul in STEP 7 konfiguriert haben. Wenn Sie die Anweisung DPRD_DAT aufrufen, können Sie nur auf die Daten eines Moduls bzw. einer DP-Kennung an der konfigurierten Anfangsadresse zugreifen.</p>
	<pre>ret_val := DPWR_DAT(     laddr:=_word_in_,     record:=_variant_in_);</pre>	<p>Mit der Anweisung DPWR_DAT übertragen Sie die Daten in RECORD konsistent in den adressierten DP-Normslave bzw. das PROFINET IO-Gerät. Der Quellbereich muss dieselbe Länge haben wie die, die Sie für das ausgewählte Modul in STEP 7 konfiguriert haben.</p>

Die CPU unterstützt bis zu 64 Bytes konsistenter Daten. Bei Bereichen konsistenter Daten größer als 64 Bytes müssen die Anweisungen DPRD\_DAT und DPWR\_DAT verwendet werden. Diese Anweisungen können bei Bedarf für Datenbereiche ab 1 Byte verwendet werden. Bei Verweigerung des Zugriffs wird der Fehlercode W#16#8090 gemeldet.

#### Hinweis

Wenn Sie die Anweisungen DPRD\_DAT und DPWR\_DAT mit konsistenten Daten verwenden, müssen Sie diese konsistenten Daten aus der automatischen Aktualisierung des Prozessabblids herausnehmen. Weitere Informationen hierzu finden Sie unter "PLC-Grundlagen: Ausführung des Anwenderprogramms" (Seite 73).

Tabelle 8- 79 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
LADDR	IN	HW_IO (Word)	<ul style="list-style-type: none"> <li>Konfigurierte Startadresse im Bereich "E" des Moduls, aus dem die Daten gelesen werden (DPRD_DAT)</li> <li>Konfigurierte Startadresse aus dem Prozessabbild der Ausgänge des Moduls, in das die Daten geschrieben werden (DPWR_DAT)</li> </ul> Die Adressen müssen im Hexadezimalformat eingegeben werden (Beispiel: eine Eingangs- oder Ausgangsadresse von 100 bedeutet: LADDR:=W#16#64).
RECORD	OUT	Variant	Zielbereich der gelesenen Anwenderdaten (DPRD_DAT) oder Quellbereich der zu schreibenden Anwenderdaten (DPWR_DAT). Dieser Bereich muss genau so groß sein, wie Sie für das ausgewählte Modul mit STEP 7 konfiguriert haben. Nur der Datentyp Byte ist zulässig.
RET_VAL	OUT	Int	Tritt ein Fehler auf, während die Funktion aktiv ist, enthält der Rückgabewert einen Fehlercode.

### Funktionsweise von DPRD\_DAT

Der Zielbereich muss dieselbe Länge haben wie die, die für das ausgewählte Modul in STEP 7 konfiguriert ist. Tritt während der Datenübertragung kein Fehler auf, werden die ausgelesenen Daten in den von RECORD angegebenen Zielbereich eingegeben.

Wenn Sie aus einem DP-Normslave mit modularem Aufbau oder mit mehreren DP-Kennungen lesen, können Sie pro Aufruf der Anweisung DPRD\_DAT nur auf die Daten eines Moduls bzw. einer DP-Kennung zugreifen, indem Sie die konfigurierte Anfangsadresse angeben.

### Funktionsweise von DPWR\_DAT

Sie übertragen die Daten in RECORD konsistent in den adressierten DP-Normslave bzw. das PROFINET IO-Gerät. Die Daten werden synchron übertragen, d.h. der Schreibvorgang wird zusammen mit der Anweisung beendet.

Der Quellbereich muss dieselbe Länge haben wie die, die Sie für das ausgewählte Modul in STEP 7 konfiguriert haben.

Wenn der DP-Normslave modular aufgebaut ist, können Sie nur auf ein Modul des DP-Slaves zugreifen.

Tabelle 8- 80 Fehlercodes von DPRD\_DAT und DPWR\_DAT

Fehlercode	Beschreibung
0000	Kein Fehler
808x	Systemfehler mit externem DP-Schnittstellenmodul
8090	Es gilt einer der folgenden Fälle: <ul style="list-style-type: none"> <li>Sie haben für die angegebene logische Basisadresse kein Modul konfiguriert.</li> <li>Sie haben die Einschränkung bezüglich der Länge konsistenter Daten ignoriert.</li> <li>Sie haben die Anfangsadresse im Parameter LADDR nicht im Hexadezimalformat eingegeben.</li> </ul>
8092	In der Referenz Any ist ein anderer Datentyp als Byte angegeben.

Fehlercode	Beschreibung
8093	An der in LADDR angegebenen logischen Adresse ist kein DP-Modul/PROFINET IO-Gerät vorhanden, aus dem konsistente Daten gelesen (DPRD_DAT) oder in das konsistente Daten geschrieben (DPWR_DAT) werden können.
80A0	Beim Zugriff auf die E/A-Geräte wurde ein Zugriffsfehler erkannt (DPRD_DAT).
80A1	Beim Zugriff auf die E/A-Geräte wurde ein Zugriffsfehler erkannt (DPWR_DAT).
80B0	Slave-Ausfall auf dem externen DP-Schnittstellenmodul
80B1	Die Länge des angegebenen Zielbereichs (DPRD_DAT) oder Quellbereichs (DPWR_DAT) ist nicht mit der in STEP 7 konfigurierten Länge der Anwenderdaten identisch.
80B2, 80B3, 80C2, 80Fx	Systemfehler mit externem DP-Schnittstellenmodul (DPRD_DAT) und (DPWR_DAT)
87xy, 808x	Systemfehler mit externem DP-Schnittstellenmodul (DPRD_DAT)
85xy	Systemfehler mit externem DP-Schnittstellenmodul (DPWR_DAT)
80C0	Die Daten wurden noch nicht vom Modul gelesen (DPRD_DAT).
80C1	Die Daten des vorherigen Schreibauftrags für das Modul wurden vom Modul noch nicht verarbeitet (DPWR_DAT).
8xyy <sup>1</sup>	Allgemeine Fehlerinformation

Weitere Informationen zu allgemeinen Fehlercodes finden Sie unter "Erweiterte Anweisungen, dezentrale E/A: Fehlerinformationen für RDREC, WRREC und RALRM" (Seite 296).

**Hinweis**

Wenn Sie auf DPV1-Slaves zugreifen, können Fehlerinformationen von diesen Slaves vom DP-Master an die Anweisung weitergeleitet werden.

**8.3.6 DPNRM\_DG**

Die Anweisung DPNRM\_DG (Diagnosedaten lesen) können Sie für PROFIBUS einsetzen.

Tabelle 8- 81 Anweisung DPNRM\_DG


KOP/FUP	SCL	Beschreibung
	<pre>ret_val := DPNRM_DG(     req:=_bool_in_,     laddr:=_word_in_,     record=&gt;_variant_out_,     busy=&gt;_bool_out_);</pre>	<p>Mit der Anweisung DPNRM_DG können Sie die aktuellen Diagnosedaten eines DP-Slaves in dem Format, das in EN 50 170 Band 2, PROFIBUS angegeben ist, lesen. Die ausgelesenen Daten werden nach einer fehlerfreien Datenübertragung in den von RECORD angegebenen Zielbereich eingegeben.</p>

Tabelle 8- 82 Datentypen für die Parameter von DPNRM\_DG

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	REQ=1: Leseanforderung
LADDR	IN	HW_DPSLAVE	Konfigurierte Diagnoseadresse des DP-Slaves: muss die Adresse der Station und nicht des E/A-Geräts sein. Um die Diagnoseadresse zu ermitteln, wählen Sie in der Netzsicht der "Gerätekonfiguration" die Station (und nicht das Abbild des Geräts).  Geben Sie die Adressen im Hexadezimalformat ein. Beispiel: Diagnoseadresse 1022 bedeutet LADDR:=W#16#3FE.
RET_VAL	OUT	Int	Tritt ein Fehler auf, während die Funktion aktiv ist, enthält der Rückgabewert einen Fehlercode. Tritt kein Fehler auf, wird die Länge der tatsächlich übertragenen Daten in RET_VAL eingegeben.
RECORD	OUT	Variant	Zielbereich für die Diagnosedaten, die gelesen wurden. Nur der Datentyp Byte ist zulässig. Die Mindestlänge des zu lesenden Datensatzes bzw. der Zielbereich ist 6. Die Höchstlänge des zu sendenden Datensatzes ist 240.  Normslaves können mehr als 240 Bytes an Diagnosedaten liefern, maximal 244 Bytes. Dann werden nur die ersten 240 Bytes in den Zielbereich übertragen, und das Überlaufbit wird in den Daten gesetzt.
BUSY	OUT	Bool	BUSY = 1: Der Leseauftrag ist noch nicht beendet

Sie starten den Leseauftrag, indem Sie dem Eingangsparameter REQ im Aufruf der Anweisung DPNRM\_DG den Wert 1 zuweisen. Der Leseauftrag wird asynchron ausgeführt, in anderen Worten, es sind dafür mehrere Aufrufe der Anweisung DPNRM\_DG erforderlich. Der Status des Auftrags wird von den Ausgangsparametern RET\_VAL und BUSY angegeben.

Tabelle 8- 83 Struktur der Slave-Diagnosedaten

Byte	Beschreibung
0	Teilnehmerstatus 1
1	Teilnehmerstatus 2
2	Teilnehmerstatus 3
3	Master-Teilnehmernummer
4	Hersteller-ID (oberes Byte)
5	Hersteller-ID (unteres Byte)
6 ...	Weitere slave-spezifische Diagnoseinformationen

Tabelle 8- 84 Fehlercodes der Anweisung DPNRM\_DG

Fehlercode	Beschreibung	Einschränkung
0000	Kein Fehler	-
7000	Erster Aufruf mit REQ = 0: Keine Datenübertragung aktiv; BUSY hat den Wert 0.	-
7001	Erster Aufruf mit REQ = 1: Keine Datenübertragung aktiv; BUSY hat den Wert 1.	Dezentrale E/A

Fehlercode	Beschreibung	Einschränkung
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Datenübertragung ist bereits aktiv; BUSY hat den Wert 1.	Dezentrale E/A
8090	Angegebene logische Basisadresse ist ungültig: Keine Basisadresse vorhanden.	-
8092	In der Referenz Any ist ein anderer Datentyp als Byte angegeben.	-
8093	<ul style="list-style-type: none"> <li>Diese Anweisung ist für das von LADDR angegebene Modul nicht zulässig (S7-DP-Module für die S7-1200 sind zulässig).</li> <li>LADDR gibt statt der Station das E/A-Gerät an. Um die Diagnoseadresse für LADDR zu ermitteln, wählen Sie in der Netzsicht der "Gerätekonfiguration" die Station (und nicht das Abbild des Geräts).</li> </ul>	-
80A2	<ul style="list-style-type: none"> <li>DP-Protokollfehler in Schicht 2 (Beispiel: Slave-Ausfall oder Busprobleme)</li> <li>Bei ET200S kann der Datensatz nicht im DPV0-Modus gelesen werden.</li> </ul>	Dezentrale E/A
80A3	DP-Protokollfehler mit User-Interface/User	Dezentrale E/A
80A4	Kommunikationsproblem auf dem Kommunikationsbus	Der Fehler tritt zwischen der CPU und dem externen DP-Schnittstellenmodul auf.
80B0	<ul style="list-style-type: none"> <li>Die Anweisung ist für den Modultyp nicht zulässig.</li> <li>Das Modul erkennt den Datensatz nicht.</li> <li>Datensatznummer 241 ist nicht zulässig.</li> </ul>	-
80B1	Die im Parameter RECORD angegebene Länge ist falsch.	Angegebene Länge > Datensatzlänge
80B2	Der konfigurierte Steckplatz ist nicht belegt.	-
80B3	Tatsächlicher Modultyp entspricht nicht dem erforderlichen Modultyp.	-
80C0	Keine Diagnoseinformationen vorhanden.	-
80C1	Die Daten des vorherigen Schreibauftrags für denselben Datensatz auf dem Modul wurden vom Modul noch nicht verarbeitet.	-
80C2	Das Modul verarbeitet derzeit die maximal mögliche Anzahl von Aufträgen für eine CPU.	-
80C3	Die erforderlichen Ressourcen (Speicher usw.) sind derzeit ausgelastet.	-
80C4	Interner temporärer Fehler. Der Auftrag konnte nicht verarbeitet werden. Wiederholen Sie den Auftrag. Tritt dieser Fehler häufig auf, prüfen Sie Ihr System auf Quellen elektrischer Störung.	-
80C5	Dezentrale E/A nicht verfügbar	Dezentrale E/A
80C6	Datensatzübertragung wurde wegen Prioritätsklasseneinordnung abgebrochen (Neustart oder Hintergrund)	Dezentrale E/A
8xyy <sup>1</sup>	Allgemeine Fehlercodes	

Weitere Informationen zu allgemeinen Fehlercodes finden Sie unter "Erweiterte Anweisungen, dezentrale E/A: Fehlerinformationen für RDREC, WRREC und RALRM" (Seite 296).



## 8.4 Alarme

### 8.4.1 Operationen ATTACH und DETACH

Mit den Anweisungen ATTACH und DETACH können Sie Unterprogramme, die durch Alarmereignisse angestoßen werden, aktivieren und deaktivieren.

Tabelle 8- 85 Anweisungen ATTACH und DETACH

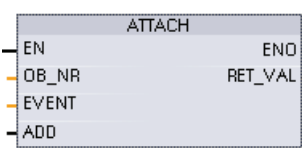
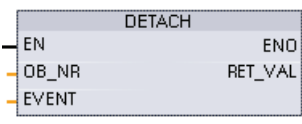
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := ATTACH(   ob_nr:=_int_in_,   event:=_event_att_in_,   add:=_bool_in_);</pre>	Die Anweisung ATTACH aktiviert die Ausführung des Unterprogramms eines Alarm-OBs bei einem Prozessereignis.
	<pre>ret_val := DETACH(   ob_nr:=_int_in_,   event:=_event_att_in_);</pre>	Die Anweisung DETACH deaktiviert die Ausführung des Unterprogramms eines Alarm-OBs bei einem Prozessereignis.

Tabelle 8- 86 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
OB_NR	IN	OB_ATT	Organisationsbausteinkennung: Wählen Sie einen der verfügbaren Prozessalarm-OBs aus, die über die Funktion "Neuen Baustein hinzufügen" angelegt wurden. Doppelklicken Sie auf das Parameterfeld und dann auf das Symbol, um die verfügbaren OBs anzuzeigen.
EVENT	IN	EVENT_ATT	Ereigniskennung: Wählen Sie eines der verfügbaren Prozessalarmereignisse aus, die in der PLC-Gerätekonfiguration für digitale Eingänge oder schnelle Zähler aktiviert wurden. Doppelklicken Sie auf das Parameterfeld und dann auf das Symbol, um die verfügbaren Ereignisse anzuzeigen.
ADD (nur ATTACH)	IN	Bool	<ul style="list-style-type: none"> <li>ADD = 0 (Standard): Dieses Ereignis ersetzt alle bisherigen Ereigniszuordnungen für diesen OB.</li> <li>ADD = 1: Dieses Ereignis wird zu allen bisherigen Ereigniszuordnungen für diesen OB hinzugefügt.</li> </ul>
RET_VAL	OUT	Int	Ausführungsbedingung

## Prozessalarmereignisse

Die folgenden Prozessalarmereignisse werden von der CPU unterstützt:

- Ereignisse Steigende Flanke (alle integrierten digitalen Eingänge der CPU und alle digitalen Eingänge des SB)
  - Eine steigende Flanke tritt auf, wenn der digitale Eingang in Reaktion auf den Signalwechsel eines an den Eingang angeschlossenen Geräts von AUS nach EIN wechselt.
- Ereignisse Fallende Flanke (alle integrierten digitalen Eingänge der CPU und alle digitalen Eingänge des SB)
  - Eine fallende Flanke tritt auf, wenn der digitale Eingang von EIN nach AUS wechselt.
- Ereignisse Aktueller Wert des schnellen Zählers (HSC) = Referenzwert (CV = RV) (HSC 1 bis 6)
  - Ein Alarm CV = RV wird für einen HSC generiert, wenn der aktuelle Zählwert von einem benachbarten Wert zu dem Wert wechselt, der genau dem zuvor eingerichteten Referenzwert entspricht.
- Ereignisse HSC-Richtungswechsel (HSC 1 bis 6)
  - Ein Richtungswechselereignis tritt auf, wenn erkannt wird, dass der HSC von Erhöhen zu Verringern oder von Verringern zu Erhöhen wechselt.
- Ereignisse Externes Rücksetzen HSC (HSC 1 bis 6)
  - Bestimmte HSC-Betriebsarten gestatten die Zuweisung eines digitalen Eingangs für externes Rücksetzen, um den Zählwert des HSC auf Null zurückzusetzen. Ein Ereignis Externes Rücksetzen tritt für einen solchen HSC auf, wenn dieser Eingang von AUS nach EIN wechselt.

## Aktivieren von Prozessalarmereignissen in der Gerätekonfiguration

Prozessalarme müssen während der Gerätekonfiguration aktiviert werden. Sie müssen das Kästchen zum Aktivieren der Ereignisse in der Gerätekonfiguration für einen digitalen Eingangskanal oder einen HSC selektieren, wenn Sie dieses Ereignis während der Konfiguration oder zur Laufzeit zuweisen möchten.

Mögliche Optionskästchen in der PLC-Gerätekonfiguration:

- Digitaleingang
  - Erkennung steigender Flanken aktivieren
  - Erkennung fallender Flanken aktivieren
- Schneller Zähler (HSC)
  - Diesen schnellen Zähler aktivieren
  - Alarm für Zählerwert gleich Referenzwert generieren
  - Alarm für Ereignis Externes Rücksetzen generieren
  - Alarm für Richtungswechselereignis generieren

## Neue Prozessalarm-OBs in Ihr Programm einfügen

Standardmäßig wird bei der ersten Aktivierung des Ereignisses dem Ereignis kein OB zugeordnet. Dies wird durch die Kennung "<nicht verbunden>" in der Gerätekonfiguration "Prozessalarm:" gekennzeichnet. Nur Prozessalarm-OBs können einem Prozessalarmereignis zugeordnet werden. Alle vorhandenen Prozessalarm-OBs werden in der Klappliste "Prozessalarm:" aufgeführt. Wird kein OB aufgeführt, müssen Sie einen OB vom Typ "Prozessalarm" wie folgt anlegen. Gehen Sie in der Projektnavigation unter "Programmbausteine" folgendermaßen vor:

1. Doppelklicken Sie auf "Neuen Baustein hinzufügen", wählen Sie "Organisationsbaustein (OB)" und dann "Prozessalarm".
2. Optional können Sie den OB umbenennen, die Programmiersprache (KOP oder FUP) festlegen und die Bausteinnummer auswählen (schalten Sie auf manuelle Bearbeitung um und wählen Sie eine andere als die vorgeschlagene Bausteinnummer aus).
3. Bearbeiten Sie den OB und fügen Sie die Reaktion des Programms ein, die bei Auftreten des Ereignisses ausgeführt werden soll. Sie können bis zu einer Schachtelungstiefe von vier Bausteinen FCs und FBs aus diesem OB aufrufen.

## Parameter OB\_NR

Die Namen aller vorhandenen Prozessalarm-OBs werden in der Klappliste "Prozessalarm:" in der Gerätekonfiguration und in der Klappliste für den Parameter "OB\_NR" der Anweisungen ATTACH /DETACH aufgeführt.

## Parameter EVENT

Wenn ein Prozessalarmereignis aktiviert wird, wird diesem bestimmten Ereignis ein eindeutiger Standardereignisname zugewiesen. Diesen Ereignisnamen können Sie durch Bearbeiten im Eingabefeld "Ereignisname:" ändern, er muss jedoch eindeutig sein. Die Ereignisnamen werden in Variablennamen in der Variablen-tabelle "Konstanten" umgewandelt und erscheinen in der Klappliste des Parameters "EVENT" in den Anweisungsfeldern ATTACH und DETACH. Der Wert dieser Variablen ist eine interne Nummer, die das Ereignis kennzeichnet.

## Allgemeine Funktionsweise

Jeder Prozessalarm kann einem Prozessalarm-OB zugeordnet werden, der bei Auftreten des Prozessalarmereignisses in die Warteschlange für die Bearbeitung gestellt wird. Die Zuordnung von OB und Ereignis kann während der Konfiguration oder zur Laufzeit erfolgen.

Sie haben die Möglichkeit, zur Zeit der Konfiguration einem aktivierten Ereignis einen OB zuzuordnen bzw. die Zuordnung aufzuheben. Um während der Konfiguration einem Ereignis einen OB zuzuordnen, wählen Sie in der Klappliste der verfügbaren Prozessalarm-OBs unter "Prozessalarm:" einen OB aus (klicken Sie auf den abwärts gerichteten Pfeil an der Klappliste). Wählen Sie in dieser Liste den Namen des gewünschten OBs aus, oder wählen Sie "<nicht verbunden>", um die Zuordnung aufzuheben.

Sie können ein aktiviertes Prozessalarmereignis auch während der Laufzeit zuordnen bzw. die Zuordnung aufheben. Mit den Anweisungen ATTACH und DETACH können Sie während der Laufzeit (bei Bedarf mehrmals) ein aktiviertes Prozessalarmereignis dem entsprechenden OB zuordnen bzw. die Zuordnung aufheben. Wenn derzeit kein OB zugeordnet ist (entweder aufgrund der Auswahl "<nicht verbunden>" in der Gerätekonfiguration oder als Ergebnis der Ausführung von DETACH), wird der aktivierte Prozessalarm ignoriert.

### Funktionsweise von DETACH

Mit der Anweisung DETACH heben Sie die Zuordnung eines bestimmten Ereignisses oder aller Ereignisse zu einem bestimmten OB auf. Ist ein EVENT angegeben, so wird nur dieses eine Ereignis von dem angegebenen Organisationsbaustein OB\_NR getrennt. Alle anderen derzeit zu diesem Organisationsbaustein OB\_NR zugeordneten Ereignisse bleiben zugeordnet. Ist kein EVENT angegeben, werden alle derzeit zu diesem Organisationsbaustein OB\_NR zugeordneten Ereignisse getrennt.

### Bedingungscode

Tabelle 8- 87 Bedingungscode

RET_VAL (W#16#...)	ENO	Beschreibung
0000	1	Kein Fehler
0001	1	Keine Zuordnung vorhanden (nur DETACH)
8090	0	OB ist nicht vorhanden
8091	0	OB ist vom falschen Typ
8093	0	Ereignis ist nicht vorhanden

## 8.4.2 Weckalarme

### 8.4.2.1 SET\_CINT (Weckalarm setzen)

Tabelle 8- 88 SET\_CINT (Weckalarm setzen)

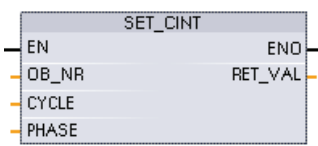
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := SET_CINT(   ob_nr:=_int_in_,   cycle:=_udint_in_,   phase:=_udint_in_);</pre>	<p>Der angegebene Alarm-OB wird gesetzt, um die zyklische Ausführung zu starten, die den Programmzyklus unterbricht.</p>

Tabelle 8- 89 Datentypen für die Parameter

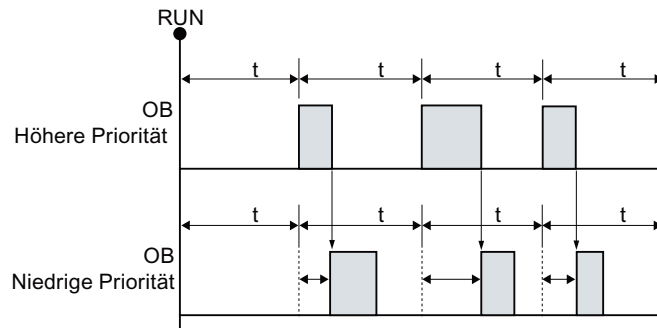
Parameter und Datentyp		Datentyp	Beschreibung
OB_NR	IN	OB_CYCLIC	OB-Nummer (symbolisch adressierbar)
CYCLE	IN	UDInt	Zeitintervall in Mikrosekunden
PHASE	IN	UDInt	Phasenverschiebung in Mikrosekunden
RET_VAL	OUT	Int	Ausführungsbedingung

Beispiele für Zeitparameter:

- Ist die CYCLE-Zeit = 100 us, dann unterbricht der von OB\_NR angegebene Alarm-OB den Programmzyklus alle 100 us. Der Alarm-OB wird ausgeführt und übergibt anschließend die Ausführungssteuerung wieder an den Programmzyklus an der Stelle, an der die Unterbrechung auftrat.
- Ist die CYCLE-Zeit = 0, dann wird das Alarmereignis deaktiviert und der Alarm-OB wird nicht ausgeführt.
- Die PHASE-Zeit (Phasenverschiebung) ist eine angegebene Verzögerungszeit, die auftritt, bevor das CYCLE-Zeitintervall beginnt. Mit der Phasenverschiebung können Sie die Ausführung von OBs mit niedrigerer Priorität zeitlich steuern.

Wenn OBs niedrigerer und höherer Priorität in demselben Zeitintervall aufgerufen werden, wird der OB niedrigerer Priorität erst aufgerufen, wenn die Bearbeitung des OBs höherer Priorität beendet ist. Der Ausführungsbeginn des OBs niedrigerer Priorität kann sich je nach Verarbeitungszeit des OBs höherer Priorität verschieben.

OB-Aufruf ohne Phasenverschiebung



Wenn Sie die Ausführung eines OBs niedrigerer Priorität mit fester Zykluszeit starten möchten, dann muss die Phasenverschiebung größer sein als die Verarbeitungszeit des OBs höherer Priorität.

OB-Aufruf mit Phasenverschiebung

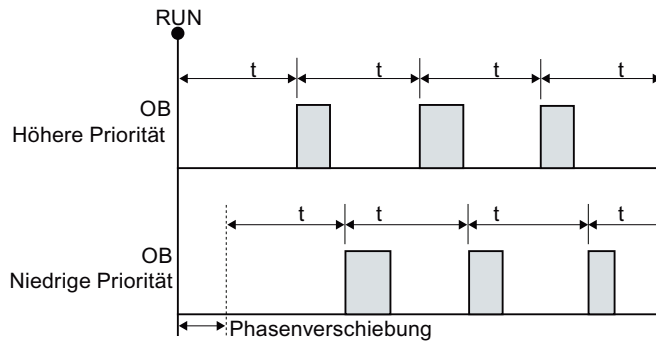


Tabelle 8- 90 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	OB ist nicht vorhanden oder hat den falschen Typ
8091	Ungültige Zykluszeit
8092	Ungültige Phasenverschiebungszeit
80B2	OB hat kein zugehöriges Ereignis

### 8.4.2.2 QRY\_CINT (Weckalarm abfragen)

Tabelle 8- 91 QRY\_CINT (Weckalarm abfragen)

KOP/FUP	SCL	Beschreibung
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <pre> QRY_CINT EN          END OB_NR      RET_VAL           CYCLE           PHASE           STATUS                     </pre> </div>	<pre> ret_val := QRY_CINT(   ob_nr:=_int_in_,   cycle=&gt;_udint_out_,   phase=&gt;_udint_out_,   status=&gt;_word_out_);                     </pre>	Parameter und Ausführungszustand eines Weckalarm-OBs werden abgerufen. Die zurückgegebenen Werte existierten zur Zeit der Ausführung von QRY_CINT.

Tabelle 8- 92 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
OB_NR	IN	OB_CYCLIC	OB-Nummer (symbolisch adressierbar, z. B. OB_MyOBName)
RET_VAL	OUT	Int	Ausführungsbedingung
CYCLE	OUT	UDInt	Zeitintervall in Mikrosekunden
PHASE	OUT	UDInt	Phasenverschiebung in Mikrosekunden
STATUS	OUT	Word	Statuscode des Weckalarms: <ul style="list-style-type: none"> <li>• Bits 0 bis 4, siehe STATUS-Tabelle unten</li> <li>• Sonstige Bits, immer 0</li> </ul>

Tabelle 8- 93 Parameter STATUS

Bit	Wert	Beschreibung
0	0	Während CPU RUN
	1	Beim Anlauf
1	0	Der Alarm ist aktiviert.
	1	Der Alarm wird über die Anweisung DIS_IRT deaktiviert.
2	0	Der Alarm ist nicht aktiv oder abgelaufen.
	1	Der Alarm ist aktiv.
4	0	Der von OB_NR angegebene OB existiert nicht.
	1	Der von OB_NR angegebene OB existiert.
Sonstige Bits		Immer 0

Tritt ein Fehler auf, zeigt RET\_VAL den entsprechenden Fehlercode und den Parameter STATUS = 0 an.

Tabelle 8- 94 Parameter RET\_VAL

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	OB ist nicht vorhanden oder hat den falschen Typ.
80B2	OB hat kein zugehöriges Ereignis.

### 8.4.3 Verzögerungsalarne

Sie können die Bearbeitung des Verzögerungsalarms mit den Anweisungen SRT\_DINT und CAN\_DINT starten und abbrechen und den Alarmzustand mit der Anweisung QRY\_DINT abfragen. Jeder Verzögerungsalarm ist ein einmaliges Ereignis, das nach einer vorgegebenen Verzögerungszeit auftritt. Wird das Zeitverzögerungsereignis gelöscht, bevor die Zeitverzögerung abgelaufen ist, tritt der Alarm nicht im Programm auf.

Tabelle 8- 95 Anweisungen SRT\_DINT, CAN\_DINT und QRY\_DINT

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := SRT_DINT(   ob_nr:=_int_in_,   dtime:=_time_in_,   sign:=_word_in_);</pre>	SRT_DINT startet einen Verzögerungsalarm, der einen OB ausführt, wenn die im Parameter DTIME festgelegte Verzögerungszeit abgelaufen ist.
	<pre>ret_val := CAN_DINT(   ob_nr:=_int_in_);</pre>	CAN_DINT löscht einen Verzögerungsalarm, der bereits gestartet ist. Der Verzögerungsalarm-OB wird dann nicht ausgeführt.
	<pre>ret_val := QRY_DINT(   ob_nr:=_int_in_,   status=&gt;_word_out_);</pre>	QRY_DINT fragt den Zustand des vom Parameter OB_NR angegebenen Verzögerungsalarms ab.

Tabelle 8- 96 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
OB_NR	IN	OB_DELAY
		Organisationsbaustein (OB), der nach einer Zeitverzögerung gestartet werden soll: Wählen Sie einen der verfügbaren Verzögerungsalarm-OBs aus, die über die Funktion "Neuen Baustein hinzufügen" in der Projektnavigation angelegt wurden. Doppelklicken Sie auf das Parameterfeld und dann auf das Symbol, um die verfügbaren OBs anzuzeigen.
DTIME <sup>1</sup>	IN	Time
		Verzögerungswert (1 bis 60000 ms)
SIGN <sup>1</sup>	IN	Word
		Von der S7-1200 nicht verwendet: Jeder Wert wird angenommen. Um Fehler zu verhindern, muss ein Wert zugewiesen werden.
RET_VAL	OUT	Int
		Ausführungsbedingung
STATUS	OUT	Word
		Anweisung QRY_DINT: Zustand des angegebenen Verzögerungsalarm-OBs, siehe Tabelle unten

<sup>1</sup> Nur bei SRT\_DINT



## Bedienung

Die Anweisung SRT\_DINT gibt eine Verzögerung an, startet die interne Zeit für die Verzögerung und weist dem Timeout-Ereignis der Zeitverzögerung das Unterprogramm eines Verzögerungsalarm-OBs zu. Wenn die angegebene Verzögerungszeit abgelaufen ist, wird eine Programmunterbrechung generiert, die die Ausführung des zugeordneten Verzögerungsalarm-OBs anstößt. Mit der Anweisung CAN\_DINT können Sie einen gestarteten Verzögerungsalarm abbrechen, bevor die angegebene Verzögerungszeit erreicht ist. Insgesamt dürfen maximal vier Verzögerungs- und Weckalarmereignisse aktiv sein.

### Unterprogramme von Verzögerungsalarm-OBs in Ihr Projekt einfügen

Den Anweisungen SRT\_DINT und CAN\_DINT können nur Verzögerungsalarm-OBs zugeordnet werden. In einem neuen Projekt ist noch kein Verzögerungsalarm-OB vorhanden. Sie müssen Verzögerungsalarm-OBs in Ihr Projekt einfügen. Um einen Verzögerungsalarm-OB anzulegen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in der Projektnavigation unter "Programmbausteine" auf "Neuen Baustein hinzufügen", wählen Sie "Organisationsbaustein (OB)" und dann "Verzögerungsalarm".
2. Sie können den OB umbenennen, die Programmiersprache festlegen und die Bausteinnummer auswählen. Schalten Sie auf manuelle Bearbeitung um, wenn Sie eine andere als die automatisch zugeordnete Bausteinnummer auswählen möchten.
3. Bearbeiten Sie das Unterprogramm des Verzögerungsalarm-OBs und fügen Sie die Reaktion des Programms ein, die bei Auftreten des Verzögerungsereignisses ausgeführt werden soll. Sie können bis zu einer Schachtelungstiefe von maximal vier Bausteinen andere FCs und FBs aus dem Verzögerungsalarm-OB aufrufen.
4. Die Namen der neu zugewiesenen Verzögerungsalarm-OBs stehen zur Verfügung, wenn Sie den Parameter OB\_NR der Anweisungen SRT\_DINT und CAN\_DINT bearbeiten.

### Parameter STATUS von QRY\_DINT

Tabelle 8- 97 Liegt ein Fehler vor (REL\_VAL <> 0), dann ist STATUS = 0.

Bit	Wert	Beschreibung
0	0	In RUN
	1	Beim Anlauf
1	0	Der Alarm ist aktiviert.
	1	Der Alarm ist deaktiviert.
2	0	Der Alarm ist nicht aktiv oder abgelaufen.
	1	Der Alarm ist aktiv.
4	0	Ein OB mit der in OB_NR angegebenen OB-Nummer existiert nicht.
	1	Ein OB mit der in OB_NR angegebenen OB-Nummer existiert.
Sonstige Bits		Immer 0

**Bedingungscode**



Tabelle 8- 98 Bedingungscode für SRT\_DINT, CAN\_DINT und QRY\_DINT

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	Falscher Parameter OB_NR
8091	Falscher Parameter DTIME
80A0	Verzögerungsalarm noch nicht gestartet.

**8.4.4 Asynchronereignisalarme**

Die Alarmbearbeitung kann mit den Anweisungen DIS\_AIRT und EN\_AIRT aktiviert und deaktiviert werden.

Tabelle 8- 99 Anweisungen DIS\_AIRT und EN\_AIRT

KOP/FUP	SCL	Beschreibung
	DIS_AIRT ( ) ;	DIS_AIRT verzögert die Bearbeitung neuer Alarmereignisse. Sie können DIS_AIRT in einem OB mehrmals ausführen.
	EN_AIRT ( ) ;	EN_AIRT aktiviert die Bearbeitung von Alarmereignissen, die zuvor mit der Anweisung DIS_AIRT deaktiviert wurden. Jede Ausführung von DIS_AIRT muss mit einer Ausführung von EN_AIRT gelöscht werden.  Die Anweisungen EN_AIRT müssen in demselben OB oder in einer aus demselben OB aufgerufenen FC bzw. FB ausgeführt werden, bevor für diesen OB erneut Alarme aktiviert werden.


 <b>WARNUNG</b>
<p>Wenn die bisherige Einstellung der Filterzeit eines Digitaleingangskanals verändert wird, muss möglicherweise bis zu 20,0 ms lang ein neuer Eingangswert mit Pegel "0" vorhanden sein, damit der Filter vollständig auf neue Eingänge reagiert. Während dieses Zeitraums werden kurze Impulse mit Pegel "0", die kürzer als 20,0 ms sind, möglicherweise nicht erkannt oder gezählt.</p> <p>Das Verändern der Filterzeiten kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.</p> <p>Um sicherzustellen, dass eine neue Filterzeit sofort wirksam wird, müssen Sie die CPU aus- und wieder einschalten.</p>

Tabelle 8- 100 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
RET_VAL	OUT	Int	Anzahl Verzögerungen = Anzahl ausgeführter DIS_AIRT in der Warteschlange.

Die Anzahl der Ausführungen von DIS\_AIRT werden vom Betriebssystem gezählt. Jede dieser Anweisungen bleibt gültig, bis sie mit einer Anweisung EN\_AIRT gelöscht wird oder bis der aktuelle OB vollständig abgearbeitet ist. Beispiel: Wenn Sie die Alarmbearbeitung fünfmal mit fünf Ausführungen von DIS\_AIRT deaktiviert haben, müssen Sie diese mit fünf Ausführungen von EN\_AIRT stornieren, damit die Alarmbearbeitung wieder aktiviert ist.

Nach der erneuten Aktivierung der Alarmereignisse werden die Alarme, die während der Ausführung von DIS\_AIRT auftraten, bearbeitet bzw. die Alarme werden bearbeitet, sobald der aktuelle OB ausgeführt wurde.

Der Parameter RET\_VAL zeigt an, wie häufig die Alarmverarbeitung deaktiviert wurde, also wie viele Ausführungen von DIS\_AIRT in die Warteschlange gestellt wurden. Die Alarmverarbeitung wird erst wieder aktiviert, wenn der Parameter RET\_VAL = 0 ist.

## 8.5 Diagnose (PROFINET oder PROFIBUS)

### 8.5.1 Diagnoseanweisungen

Die folgenden Diagnoseanweisungen können entweder mit PROFINET oder PROFIBUS verwendet werden:

- Anweisung GET\_DIAG (Seite 321): Sie können die Diagnoseinformationen aus einem angegebenen Gerät auslesen.
- Anweisung DeviceStates (Seite 318): Sie können die Betriebszustände dezentraler Peripheriegeräte in einem E/A-Untersystem abrufen.
- Anweisung ModuleStates (Seite 319): Sie können die Betriebszustände der Module in einem dezentralen Peripheriegerät abrufen.
- Anweisung LED (Seite 316): Sie können den Zustand der LEDS eines dezentralen Peripheriegeräts auslesen.

### 8.5.2 Diagnoseereignisse für die dezentrale Peripherie

---

#### Hinweis

In einem PROFIBUS IO-System geht die CPU nach einem Ladevorgang oder dem Aus- und Wiedereinschalten in den Betriebszustand RUN, sofern die Hardwarekompatibilität nicht so eingestellt ist, dass akzeptable Ersatzmodule (Seite 133) zulässig sind und mindestens ein Modul fehlt oder es sich dabei nicht um ein akzeptables Ersatzmodul für das konfigurierte Modul handelt.

---

Wie Sie in der folgenden Tabelle sehen, unterstützt die CPU diejenige Diagnose, die für die Komponenten des dezentralen E/A-Systems konfiguriert werden kann. Jeder dieser Fehler erzeugt einen Protokolleintrag im Diagnosepuffer.

Tabelle 8- 101 Behandlung von Diagnoseereignissen bei PROFINET und PROFIBUS

Fehlertyp	Diagnoseinformation für die Station?	Eintrag im Diagnosepuffer?	CPU-Betriebszustand
Diagnosefehler	Ja	Ja	Bleibt im Betriebszustand RUN
Fehler bei Baugruppenträger oder Station	Ja	Ja	Bleibt im Betriebszustand RUN
Peripheriezugriffsfehler <sup>1</sup>	Nein	Ja	Bleibt im Betriebszustand RUN
Peripheriezugriffsfehler <sup>2</sup>	Nein	Ja	Bleibt im Betriebszustand RUN
Ziehen/Stecken-Ereignis	Ja	Ja	Bleibt im Betriebszustand RUN

- <sup>1</sup> Beispielursache für einen Peripheriezugriffsfehler: Ein Modul wurde entfernt.
- <sup>2</sup> Beispielursache für einen Peripheriezugriffsfehler: Azyklische Kommunikation mit einem Submodul, das nicht kommuniziert.

Über die Anweisung GET\_DIAG (Seite 321) können Sie für jede Station die Diagnoseinformationen abrufen. Auf diese Weise können Sie die im Gerät aufgetretenen Fehler programmatisch behandeln und, sofern gewünscht, die CPU in den Betriebszustand STOP versetzen. Für dieses Verfahren müssen Sie das Hardwaregerät angeben, aus dem die Statusinformationen ausgelesen werden sollen.

Die Anweisung GET\_DIAG verwendet die "L-Adresse" (LADDR) der Station, um den Zustand der gesamten Station abzurufen. Diese L-Adresse finden Sie in der Netzsicht der Gerätekonfiguration, indem Sie den gesamten Baugruppenträger der Station auswählen (den gesamten grauen Bereich). Die L-Adresse wird in der Registerkarte "Eigenschaften" der Station angezeigt. Sie finden den Parameter LADDR für jedes einzelne Modul entweder in den Eigenschaften des Moduls (in der Gerätekonfiguration) oder in der Standardvariablen-tabelle der CPU.

### 8.5.3 Anweisung LED

Tabelle 8- 102 Anweisung LED

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := LED(     laddr:=_word_in_,     LED:=_uint_in_);</pre>	<p>Mit der Anweisung LED lesen Sie den Zustand der LEDs auf einer CPU oder Schnittstelle aus. Der angegebene LED-Zustand wird vom Ausgang RET_VAL zurückgegeben.</p>

Tabelle 8- 103 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung		
LADDR	IN	HW_IO	Identifikationsnummer der CPU oder Schnittstelle <sup>1</sup>		
LED	IN	UInt	LED-Kennnummer		
			1	RUN/STOP	Farbe 1 = grün, Farbe 2 = gelb
			2	Fehler	Farbe 1 = rot
			3	Wartung	Farbe 1 = gelb
			4	Redundanz	-/-
			5	Verbindung	Farbe 1 = grün
6	Tx/Rx	Farbe 1 = gelb			
RET_VAL	OUT	Int	Zustand der LED		

<sup>1</sup> Sie können beispielsweise die CPU (wie "PLC\_1") oder die PROFINET-Schnittstelle in der Klappliste des Parameters auswählen.

Tabelle 8- 104 Zustand von RET\_VAL

RET_VAL (W#16#...)	Beschreibung	
LED-Zustand 0 bis 9	0	LED ist nicht vorhanden
	1	Aus
	2	Farbe 1 ein (leuchtet ständig)
	3	Farbe 2 ein (leuchtet ständig)
	4	Farbe 1 blinkt mit 2 Hz
	5	Farbe 2 blinkt mit 2 Hz
	6	Farben 1 & 2 blinken abwechselnd mit 2 Hz
	7	Farbe 1 ein (Tx/Rx)
	8	Farbe 2 ein (Tx/Rx)
9	Zustand der LED ist nicht verfügbar	
8091	Von LADDR angegebenes Gerät ist nicht vorhanden	
8092	Von LADDR angegebenes Gerät unterstützt keine LEDs	
8093	LED-Kennung nicht definiert	
80Bx	Von LADDR angegebene CPU unterstützt die Anweisung LED nicht	

### 8.5.4 Anweisung DeviceStates

Tabelle 8- 105 Anweisung DeviceStates

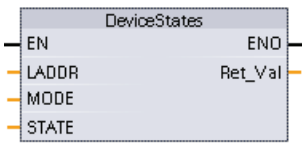
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := DeviceStates (     laddr:=hw_io_in_,     mode:=_uint_in_,     state:=_variant_inout_);</pre>	DeviceStates ruft die Betriebszustände von E/A-Geräten in einem E/A-Untersystem ab. Nach der Ausführung enthält der Parameter STATE den Fehlerzustand jedes einzelnen E/A-Geräts in einer Bitliste (für die zugewiesenen LADDR und MODE). Diese Informationen entsprechen dem Gerätestatus in der Diagnoseansicht von STEP 7.

Tabelle 8- 106 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
LADDR	IN	HW_IOSYSTEM	Logische Adresse: (Kennung des E/A-Systems)
MODE	IN	UInt	Statustyp: <ul style="list-style-type: none"> <li>• 1: Die Konfiguration des Geräts ist aktiv oder noch nicht beendet.</li> <li>• 2: Gerät defekt</li> <li>• 3: Gerät deaktiviert</li> <li>• 4: Gerät vorhanden</li> </ul>
RET_VAL	OUT	Int	Ausführungsbedingung
STATE <sup>1</sup>	InOut	Variant	Puffer, der den Fehlerzustand der einzelnen Geräte empfängt: Als Datentyp für den Parameter STATE können Sie einen beliebigen Bittyp (Bool, Byte, Word oder DWord) oder ein Array eines Bittyps auswählen. <ul style="list-style-type: none"> <li>• Zusammenfassungsbit: Bit 0 = 1, wenn eines der Zustandsbits der I/O-Geräte 1 ist</li> <li>• Zustandsbit: Zustand des E/A-Geräts mit der Stationsnummer n entsprechend dem ausgewählten MODE. Ist z. B. MODE = 2 und Bit 3 = 1 bedeutet dies, dass Station 3 fehlerhaft ist.</li> </ul>

<sup>1</sup> Bei PROFIBUS DP beträgt die Länge der Zustandsinformationen 128 Bits. Bei PROFINET I/O beträgt die Länge 1024 Bit.

Nach der Ausführung enthält der Parameter STATE den Fehlerzustand jedes einzelnen E/A-Geräts als Bitliste (für die zugewiesenen LADDR und MODE).

Tabelle 8- 107 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0	Kein Fehler
8091	LADDR ist nicht vorhanden.
8092	LADDR spricht kein E/A-System an.
8093	Ungültiger Datentyp für den Parameter STATE: Gültige Datentypen sind (Bool, Byte, Word oder Dword) oder ein Array aus (Bool, Byte, Word oder Dword)
80Bx	Die Anweisung DeviceStates wird von der CPU für diesen LADDR nicht unterstützt.
8452	Die vollständigen Zustandsdaten sind für den zugewiesenen Parameter STATE zu groß. Der Puffer STATE enthält ein Teilergebnis.

## 8.5.5 Anweisung ModuleStates

Tabelle 8- 108 Anweisung ModuleStates

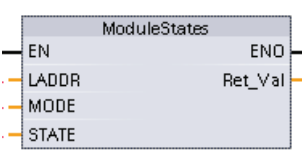
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := ModuleStates(     laddr:=_word_in_,     mode:=_uint_in_,     state:=_variant_inout);</pre>	<p>ModuleStates ruft die Betriebszustände von E/A-Modulen ab. Nach der Ausführung enthält der Parameter STATE den Fehlerzustand jedes einzelnen E/A-Moduls in einer Bitliste (für die zugewiesenen LADDR und MODE). Diese Informationen entsprechen dem Modulstatus in der Diagnoseansicht von STEP 7.</p>

Tabelle 8- 109 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
LADDR	IN	HW_DEVICE
MODE	IN	UInt
		Statustyp: <ul style="list-style-type: none"> <li>• 1: Die Konfiguration des Moduls ist aktiv oder noch nicht beendet.</li> <li>• 2: Modul defekt</li> <li>• 3: Modul deaktiviert</li> <li>• 4: Modul vorhanden</li> </ul>

Parameter und Datentyp		Datentyp	Beschreibung
RET_VAL	OUT	Int	Status (Bedingungscode)
STATE <sup>1</sup>	InOut	Variant	<p>Puffer, der den Fehlerzustand der einzelnen Module empfängt: Als Datentyp für den Parameter STATE können Sie einen beliebigen Bittyp (Bool, Byte, Word oder DWord) oder ein Array eines Bittyps auswählen.</p> <ul style="list-style-type: none"> <li>• Zusammenfassungsbit: Bit 0 = 1, wenn eines der Zustandsbits des E/A-Moduls 1 ist</li> <li>• Zustandsbit: Zustand des E/A-Moduls mit der Steckplatznummer <i>n</i> entsprechend dem ausgewählten MODE. Ist z. B. MODE = 2 und Bit 3 = 1 bedeutet dies, dass Station 3 fehlerhaft ist.</li> </ul>

<sup>1</sup> Es können maximal 128 Bit zugewiesen werden. Die Anzahl der erforderlichen Bits ist von der Nutzung Ihres E/A-Moduls abhängig.

Tabelle 8- 110 Bedingungscode

RET_VAL ( W#16#...)	Beschreibung
0	Kein Fehler
8091	Von LADDR angegebenes Modul ist nicht vorhanden.
8092	Das von LADDR angegebene Modul spricht kein E/A-Gerät an.
8093	Ungültiger Datentyp für den Parameter STATE: Gültige Datentypen sind (Bool, Byte, Word oder Dword) oder ein Array aus (Bool, Byte, Word oder Dword).
80Bx	Die Anweisung ModuleStates wird von dieser CPU für diesen LADDR nicht unterstützt.
8452	Die vollständigen Zustandsdaten sind für den zugewiesenen Parameter STATE zu groß. Der Puffer STATE enthält ein Teilergebnis.

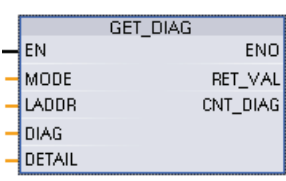


## 8.5.6 Anweisung GET\_DIAG

### Beschreibung

Mit der Anweisung GET\_DIAG können Sie die Diagnoseinformationen eines Hardwareobjekts auslesen. Das Hardwareobjekt wird über den Parameter LADDR ausgewählt. Mit dem Parameter MODE wählen Sie aus, welche Diagnoseinformationen ausgelesen werden sollen.

Tabelle 8- 111 Anweisung GET\_DIAG

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := GET_DIAG(     mode:=_uint_in_,     laddr:=_word_in_,     cnt_diag=&gt;_uint_out_,     diag=_variant_inout_,     detail:=_variant_inout_);</pre>	Liest die Diagnoseinformationen aus einem angegebenen Hardwaregerät aus.

### Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GET\_DIAG:

Tabelle 8- 112 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
MODE	IN	UInt
LADDR	IN	HW_ANY (Word)
RET_VAL	OUT	Int
CNT_DIAG	OUT	UInt
DIAG	InOut	Variant
DETAILS	InOut	Variant

**Parameter MODE**

Abhängig vom Wert des Parameters MODE werden unterschiedliche Diagnosedaten an den Ausgabeparametern DIAG, CNT\_DIAG und DETAILS ausgegeben:

Tabelle 8- 113 Parameter MODE

MODE	Beschreibung	DIAG	CNT_DIAG	DETAILS
0	Ausgabe aller unterstützten Diagnoseinformationen für ein Modul als DWord, wobei Bit X=1 kenntlich macht, dass der Modus X unterstützt wird.	Bitzeichenfolge der unterstützten Modi als DWord, wobei Bit X=1 kenntlich macht, dass der Modus X unterstützt wird.	0	-
1	Ausgabe des zugehörigen Zustands des adressierten Hardwareobjekts.	Diagnosezustand: Ausgabe in Übereinstimmung mit der Struktur DIS. (Hinweis: Beachten Sie auch die Informationen zur "DIS-Struktur" unten und das Beispiel für die Anweisung GET_DIAG am Ende des Abschnitts.)	0	-
2	Ausgabe des Zustands aller untergeordneten Module des adressierten Hardwareobjekts.	Ausgabe der Diagnosedaten in Übereinstimmung mit der Struktur DNN. (Hinweis: Beachten Sie auch die Informationen zur "DNN-Struktur" unten und das Beispiel für die Anweisung GET_DIAG am Ende des Abschnitts.)	0	Modulzustandsinformationen in Übereinstimmung mit der Struktur DiagnosticsDetails.

**DIS-Struktur**

Bei einem MODE-Parameter = 1 werden die Diagnoseinformationen in Übereinstimmung mit der Struktur DIS ausgegeben. Die folgende Tabelle zeigt die Bedeutung der einzelnen Parameterwerte:

Tabelle 8- 114 Struktur der Diagnoseinformationsquelle (DIS, Diagnostic Information Source)

Parameter	Datentyp	Wert	Beschreibung
MaintenanceState	DWord	Enum	
		0	Keine Wartung erforderlich
		1	Das Modul oder Gerät ist deaktiviert.
		2	-
		3	-
		4	-
		5	Wartung notwendig
		6	Wartung angefordert
		7	Fehler

Parameter	Datentyp	Wert	Beschreibung
		8	Zustand unbekannt/Fehler im untergeordneten Modul
		9	-
		10	Eingänge/Ausgänge sind nicht verfügbar.
Componentstate Detail	DWord	Bit-Array	Zustand der Submodule des Moduls: <ul style="list-style-type: none"> <li>• Bit 0 bis 15: Statusmeldung des Moduls</li> <li>• Bit 16 bis 31: Statusmeldung der CPU</li> </ul>
		0 bis 2 (Enum)	Weitere Informationen: <ul style="list-style-type: none"> <li>• Bit 0: Keine weiteren Informationen</li> <li>• Bit 1: Übertragung nicht zulässig</li> </ul>
		3	Bit 3 = 1: Mindestens ein Kanal unterstützt Kennzeichner für die Diagnose.
		4	Bit 4 = 1: Mindestens ein Kanal oder eine Komponente muss gewartet werden
		5	Bit 5 = 1: Für mindestens einen Kanal oder eine Komponente wurde die Wartung angefordert
		6	Bit 6 = 1: Fehler bei mindestens einem Kanal oder einer Komponente
		7 bis 10	-
		11 bis 14	Bit 11 = 1: PNIO - Submodul korrekt Bit 12 = 1: PNIO - Ersatzmodul Bit 13 = 1: PNIO - falsches Modul Bit 14 = 1: PNIO - Modul getrennt
		15	-
		16 bis 31	Von der CPU erzeugte Zustandsinformationen für Module: Bit 16 = 1: Modul deaktiviert Bit 17 = 1: CiR-Operation aktiv Bit 18 = 1: Eingang nicht verfügbar Bit 19 = 1: Ausgang nicht verfügbar Bit 20 = 1: Überlauf Diagnosepuffer Bit 21 = 1: Diagnose nicht verfügbar Bit 22 - 31: Reserviert (immer 0)
OwnState	UInt16	Enum	Der Wert des Parameters OwnState beschreibt den Wartungszustand des Moduls.
		0	Keine Störung
		1	Das Modul oder Gerät ist deaktiviert.
		2	Wartung notwendig
		3	Wartung angefordert
		4	Fehler
		5	Das Modul oder Gerät kann von der CPU nicht erreicht werden (gilt für Module und Geräte unter einer CPU).
		6	Eingänge/Ausgänge sind nicht verfügbar.
7	-		
IO State	UInt16	Bit-Array	E/A-Zustand des Moduls
		0	Bit 0 = 1: Keine Wartung erforderlich

Parameter	Datentyp	Wert	Beschreibung
		1	Bit 1 = 1: Das Modul oder Gerät ist deaktiviert.
		2	Bit 2 = 1: Wartung notwendig
		3	Bit 3 = 1: Wartung angefordert
		4	Bit 4 = 1: Fehler
		5	Bit 5 = 1: Das Modul oder Gerät kann von der CPU nicht erreicht werden (gilt für Module und Geräte unter einer CPU).
		6	Eingänge/Ausgänge sind nicht verfügbar.
		7	Kennzeichner; Bit 7 = 1, wenn Bit 0, 2 oder 3 gesetzt ist
		8 bis 15	Reserviert (immer = 0)
OperatingState	UInt16	Enum	
		0	-
		1	In STOP/Firmware-Update
		2	In STOP/Urlöschen des Speichers
		3	In STOP/Selbstanlauf
		4	In STOP
		5	Urlöschen des Speichers
		6	In START
		7	In RUN
		8	-
		9	In HOLD
		10	-
		11	-
		12	Modul defekt
		13	-
		14	Keine Spannung
		15	CiR
		16	In STOP/ohne DIS
		17	Eingang
		18	
19			
20			

**DiagnosticsDetail-Struktur**

Bei einem MODE-Parameter = 2 werden die Diagnoseinformationsdetails in Übereinstimmung mit der Struktur DiagnosticsDetail ausgegeben. Die folgende Tabelle zeigt die Bedeutung der einzelnen Parameterwerte:

Tabelle 8- 115 Struktur von DiagnosticsDetail

Parameter	Datentyp	Beschreibung
ChannelNumber	UInt	Kanalnummer
Properties	Word	

Parameter	Datentyp	Beschreibung
ALID	UInt	Identifizierungskennung des Alarms
Qualifier	DWord	Kennzeichner der Diagnosedaten
ErrorType	UDInt	Kanalfehlertyp
ExtErrorType	UDInt	Erweiterter Kanalfehlertyp
AddValue_1	UInt	Zusätzlicher Wert
AddValue_2	UInt	Zusätzlicher Wert
AddValue_3	UInt	Zusätzlicher Wert
AddValue_4	UInt	Zusätzlicher Wert

## DNN-Struktur

Bei einem MODE-Parameter = 2 werden die Diagnoseinformationsdetails in Übereinstimmung mit der Struktur DNN ausgegeben. Die folgende Tabelle zeigt die Bedeutung der einzelnen Parameterwerte:

Tabelle 8- 116 Struktur des Diagnosenavigationsknotens (DNN, Diagnostic Navigation Node)

Parameter	Datentyp	Wert	Beschreibung
SubordinateState	UINT	Enum	Zustand des untergeordneten Moduls (siehe Parameter OwnState der Struktur DIS)
SubordinateIOState	WORD	Bitarray	Zustand der Eingänge und Ausgänge des untergeordneten Moduls (siehe Parameter IO State der Struktur DIS)
DNNmode	WORD	Bitarray	<ul style="list-style-type: none"> <li>• Bit 0 = 0: Diagnose aktiviert</li> <li>• Bit 0 = 1: Diagnose deaktiviert</li> <li>• Bit 1 bis 15: Reserviert</li> </ul>

## Parameter RET\_VAL

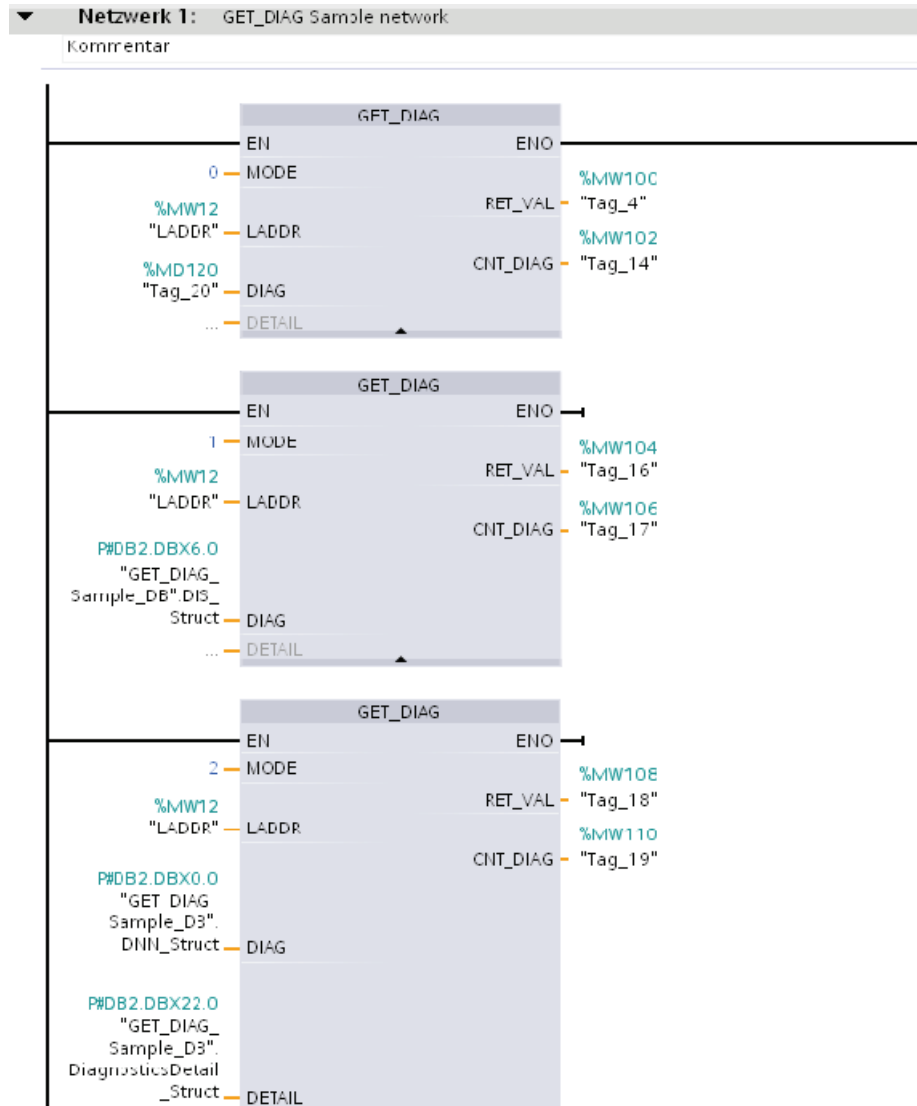
Tabelle 8- 117 Fehlercodes des Parameters RET\_VAL

Fehlercode (W#16#...)	Beschreibung
0	Kein Fehler
n	Der Datenbereich im Parameter DETAILS ist zu klein. Nicht alle Details der Diagnosedaten können ausgegeben werden.
8080	Wert im Parameter MODE wird nicht unterstützt.
8081	Typ im Parameter DIAG wird nicht mit dem ausgewählten Modus unterstützt (Parameter MODE).
8082	Typ im Parameter DETAILS wird nicht mit dem ausgewählten Modus unterstützt (Parameter MODE).
8090	LADDR ist nicht vorhanden.
8091	Der ausgewählte Kanal im Parameter CHANNEL ist nicht vorhanden.
80C1	Nicht genügend Ressourcen für die parallele Ausführung

Beispiel

Das folgende KOP-Netzwerk und der folgende DB zeigen, wie Sie die drei Betriebsarten mit den drei Strukturen verwenden:

- DIS
- DiagnosticsDetail
- DNN



	Name	Datentyp	Offset	Startwert	Rem...	Sichtbar in ...	Kommentar
1	Static						
2	DNN_Struct	DNN	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	SubordinateState	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	SubordinateIOState	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	DNNmode	Word	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	DIS_Struct	DIS	6.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	MaintenanceState	DWord	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
8	ComponentStateDetail	DWord	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
9	OwnState	UInt	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	IOState	Word	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
11	OperatingState	UInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
12	DiagnosticsDetail_Struct	DiagnosticsDetail	22.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
13	ChannelNumber	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Channel number
14	Properties	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
15	ALID	UInt	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm identifier
16	Qualifier	DWord	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
17	ErrorType	UInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Channel error type
18	ExtErrorType	UInt	16.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	extended channel error type
19	AddValue_1	UInt	20.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
20	AddValue_2	UInt	22.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
21	AddValue_3	UInt	24.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
22	AddValue_4	UInt	26.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value

- ① DNN
- ② DIS
- ③ DiagnosticsDetail

---

### Hinweis

Im DB müssen Sie den Datentyp manuell eingeben, um auf jede der drei Strukturen zuzugreifen. Eine Auswahl über eine Klappliste ist nicht möglich. Geben Sie die Datentypen genau wie im Folgenden gezeigt ein:

- DNN
  - DIS
  - DiagnosticsDetail
-

## 8.6 Impuls

### 8.6.1 Operation CTRL\_PWM

Tabelle 8- 118 Anweisung CTRL\_PWM (Impulsdauermodulation)

KOP/FUP	SCL	Beschreibung
	<pre>"CTRL_PWM_DB" (     PWM:=_word_in_,     enable:=_bool_in_,     busy=&gt;_bool_out_,     status=&gt;_word_out_);</pre>	Bietet eine feste Zykluszeit mit variabler relativer Einschaltdauer. Der PWM-Ausgang läuft nach dem Start kontinuierlich mit der angegebenen Frequenz (Zykluszeit). Die Impulsdauer wird nach Bedarf verändert, um die gewünschte Steuerung zu erzielen.

1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

2 Im SCL-Beispiel ist "CTRL\_PWM\_DB" der Name des Instanz-DBs.

#### CTRL\_HSC

Tabelle 8- 119 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
PWM	IN	HW_PWM (Word) PWM-Kennung: Die Namen aktivierter Impulsgeneratoren werden zu Variablen in der Variablen-tabelle "Konstanten" und können als PWM-Parameter genutzt werden. (Standardwert: 0)
ENABLE	IN	Bool 1 = Impulsgenerator starten 0 = Impulsgenerator stoppen
BUSY	OUT	Bool Funktion beschäftigt (Standardwert: 0)
STATUS	OUT	Word Ausführungsbedingung (Standardwert: 0)

Die Anweisung CTRL\_PWM speichert die Parameterinformationen im DB. Die Parameter des Datenbausteins werden nicht separat vom Anwender geändert, sondern von der Anweisung CTRL\_PWM gesteuert.

Geben Sie den gewünschten Impulsgenerator über den Variablennamen für den Parameter PWM an.

Wenn der Eingang EN gleich WAHR ist, startet oder stoppt die Anweisung PWM\_CTRL die angegebene PWM anhand des Werts am Eingang ENABLE. Die Impulsdauer wird von dem Wert der zugewiesenen Adresse des Ausgangsworts angegeben.

Weil die CPU die Anforderung verarbeitet, während die Anweisung CTRL\_PWM ausgeführt wird, meldet der Parameter BUSY immer FALSCH. Wird ein Fehler erkannt, wird ENO auf FALSCH gesetzt und der Parameter STATUS enthält einen Fehlercode.



Die Impulsdauer wird beim ersten Wechsel der CPU in RUN auf den in der Gerätekonfiguration eingegebenen Anfangswert gesetzt. Um die Impulsdauer zu ändern, schreiben Sie die gewünschten Werte in die in der Gerätekonfiguration angegebene Ausgangswortadresse ("Ausgangsadressen"/"Anfangsadresse:"). Um die gewünschte Impulsdauer in das entsprechende Ausgangswort zu schreiben, nutzen Sie eine Anweisung wie Übertragen, Umwandeln, Arithmetik oder PID. Sie müssen den gültigen Bereich für den Wert des Ausgangsworts einhalten (Prozent, Tausendstel, Zehntausendstel oder S7-Analogformat).

#### Hinweis

**Zu PWM und PTO zugewiesene digitale E/A können nicht geforct werden.**

Die von der Impulsdauermodulation (PWM) und der Impulsfolge (PTO) verwendeten digitalen E/A werden während der Gerätekonfiguration zugewiesen. Wenn diesen Funktionen digitale E/A zugewiesen werden, können die Werte der Adressen der zugewiesenen E/A nicht durch die Funktion zum Forcen in der Beobachtungstabelle geändert werden.

Tabelle 8- 120 Wert des Parameters STATUS

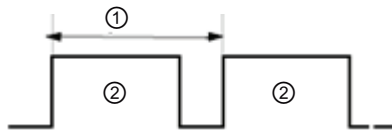
STATUS	Beschreibung
0	Kein Fehler
80A1	PWM-Kennung adressiert keine gültige PWM.

Tabelle 8- 121 Gemeinsame Bedingungscode

Bedingungscode <sup>1</sup>	Beschreibung
8022	Bereich für Eingang zu klein
8023	Bereich für Ausgang zu klein
8024	Unzulässiger Eingangsbereich
8025	Unzulässiger Ausgangsbereich
8028	Unzulässige Eingangsbitzuweisung
8029	Unzulässige Ausgangsbitzuweisung
8030	Ausgangsbereich ist ein schreibgeschützter DB.
803A	DB ist nicht vorhanden.

<sup>1</sup> Wenn einer dieser Fehler während der Ausführung eines Codebausteins auftritt, geht die CPU in den Betriebszustand STOP, es sei denn, Sie nutzen eine der Anweisungen GetError oder GetErrorID in dem Codebaustein, um eine programmierte Reaktion auf den Fehler zu erreichen.

### 8.6.2 Funktionsweise der Impulsausgänge



- ① Zykluszeit
- ② Impulsdauer

Die Impulsdauer kann als Hundertstel der Zykluszeit (0 bis 100), als Tausendstel (0 bis 1000), als Zehntausendstel (0 bis 10000) oder als S7-Analogformat angegeben werden.

Die Impulsdauer kann zwischen 0 (kein Impuls, immer aus) und Vollausschlag (kein Impuls, immer ein) liegen.

Weil der PWM-Ausgang zwischen 0 und Vollausschlag liegen kann, bietet er einen digitalen Ausgang, der in vielerlei Hinsicht einem Analogausgang gleicht. Der PWM-Ausgang kann z. B. zur Steuerung der Drehzahl eines Motors vom Stillstand bis zur vollen Drehzahl dienen oder er kann dafür eingesetzt werden, die Position eines Ventils von geschlossen bis vollständig geöffnet zu steuern.

Zur Steuerung schneller Impulsausgänge stehen zwei Impulsgeneratoren zur Verfügung: PWM und Impulsfolge (PTO). PTO wird von den Bewegungssteuerungsanweisungen genutzt. Sie können jeden Impulsgenerator entweder PWM oder PTO zuordnen, jedoch nicht beiden gleichzeitig.

Die zwei Impulsgeneratoren sind spezifischen digitalen Ausgängen zugeordnet (siehe folgende Tabelle). Sie können integrierte CPU-Ausgänge oder die Ausgänge eines optionalen Signalboards nutzen. In der folgenden Tabelle werden die Adressen der Ausgänge aufgeführt (wobei die Standardkonfiguration der Ausgänge vorausgesetzt wird). Wenn Sie die Adressen der Ausgänge geändert haben, entsprechen die Adressen den von Ihnen zugewiesenen. Unabhängig davon nutzt PTO1/PWM1 die ersten beiden digitalen Ausgänge und PTO2/PWM2 nutzt die nächsten beiden digitalen Ausgänge, entweder auf der CPU oder dem gesteckten Signalboard. Beachten Sie, dass PWM nur einen Ausgang benötigt, während PTO optional zwei Ausgänge je Kanal nutzen kann. Wenn ein Ausgang für eine Impulsfunktion nicht erforderlich ist, steht er zu anderen Zwecken zur Verfügung.

**ACHTUNG**

**Impulsfolgen können von anderen Anweisungen im Anwenderprogramm nicht verwendet werden**

Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM oder Bewegungssteuerungsanweisungen) konfigurieren, werden die entsprechenden Adressen der Ausgänge (A0.0, A0.1, A4.0 und A4.1) aus dem Speicher der Ausgänge entfernt und können nicht für andere Zwecke in Ihrem Anwenderprogramm verwendet werden. Wenn Ihr Anwenderprogramm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.

Tabelle 8- 122 Standard-Ausgangszuweisungen der Impulsgeneratoren

Beschreibung	Impuls	Richtung
PTO 0		
Integrierte E/A	A0.0	A0.1
SB-E/A	A4.0	A4.1
PWM 0		
Integrierte Ausgänge	A0.0	-
SB-Ausgänge	A4.0	-
PTO 1		
Integrierte E/A	A0.2	A0.3
SB-E/A	A4.2	A4.3
PWM 1		
Integrierte Ausgänge	A0.2	-
SB-Ausgänge	A4.2	-
PTO 2		
Integrierte E/A	A0.4 <sup>1</sup>	A0.5 <sup>1</sup>
SB-E/A	A4.0	A4.1
PWM 2		
Integrierte Ausgänge	A0.4 <sup>1</sup>	-
SB-Ausgänge	A4.1	-
PTO 3		
Integrierte E/A	A0.6 <sup>2</sup>	A0.7 <sup>2</sup>
SB-E/A	A4.2	A4.3
PWM 3		
Integrierte Ausgänge	A0.6 <sup>2</sup>	-
SB-Ausgänge	A4.3	-

- <sup>1</sup> Die CPU 1211C hat keine Ausgänge A0.4, A0.5, A0.6 und A0.7. Deshalb können diese Ausgänge bei der CPU 1211C nicht verwendet werden.
- <sup>2</sup> Die CPU 1212C hat keine Ausgänge A0.6 und A0.7. Deshalb können diese Ausgänge bei der CPU 1212C nicht verwendet werden.
- <sup>3</sup> Diese Tabelle gilt für die PTO/PWM-Funktionen der CPU 1211C, CPU 1212C, CPU 1214C und CPU 1215C.

### 8.6.3 Konfigurieren eines Impulskanals für PWM

Um die Impulsdauermodulation vorzubereiten, konfigurieren Sie zunächst einen Impulskanal in der Gerätekonfiguration durch Auswahl der CPU, des Impulsgenerators (PTO/PWM) und PWM1 oder PWM2. Aktivieren Sie den Impulsgenerator (Optionskästchen). Wenn ein Impulsgenerator aktiviert wird, wird diesem bestimmten Impulsgenerator ein eindeutiger Standardname zugewiesen. Diesen Namen können Sie durch Bearbeiten im Eingabefeld "Name:" ändern, er muss jedoch eindeutig sein. Die Namen aktivierter Impulsgeneratoren werden zu Variablen in der Variablen-Tabelle "Konstanten" und können als PWM-Parameter der Anweisung CTRL\_PWM genutzt werden.

#### ACHTUNG

Die maximale Impulsfrequenz der Impulsgeneratoren für den Digitalausgang beträgt 100 kHz (bei der CPU), 20 kHz (bei einem SB) bzw. 200 kHz (bei einem High-Speed-SB). Sie werden von STEP 7 jedoch nicht gewarnt, wenn Sie eine Achse mit einer maximalen Geschwindigkeit oder Frequenz, die diese Hardwareeinschränkung überschreitet, konfigurieren. Dies kann zu Problemen in Ihrer Anwendung führen. Stellen Sie also stets sicher, dass Sie die maximale Impulsfrequenz der Hardware nicht überschreiten.

Sie können den Impulsgenerator umbenennen, einen Kommentar einfügen und Parameter zuweisen. Dazu gehen Sie wie folgt vor:

- Der Impulsgenerator wird wie folgt genutzt: PWM oder PTO (wählen Sie PWM)
- Ausgangsquelle: integrierte CPU oder SB
- Zeitbasis: Millesekunden oder Mikrosekunden
- Impulsdauerformat:
  - Hundertstel (0 bis 100)
  - Tausendstel (0 bis 1000)
  - Zehntausendstel (0 bis 10000)
  - S7-Analogformat (0 bis 27648)
- Zykluszeit: Geben Sie Ihre Zykluszeit ein. Dieser Wert kann nur in der Gerätekonfiguration geändert werden.
- Anfangsimpulsdauer: Geben Sie die Anfangsimpulsdauer ein. Die Impulsdauer kann während der Laufzeit geändert werden.

Geben Sie die Anfangsadresse ein, um die Ausgangsadressen zu konfigurieren. Geben Sie die Adresse des Ausgangsworts ein, wo Sie den Wert für die Impulsdauer ablegen möchten.

<b>ACHTUNG</b>
<b>Impulsfolgen können von anderen Anweisungen im Anwenderprogramm nicht verwendet werden</b>
Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM oder Bewegungssteuerungsanweisungen) konfigurieren, werden die entsprechenden Adressen der Ausgänge (A0.0, A0.1, A4.0 und A4.1) aus dem Speicher der Ausgänge entfernt und können nicht für andere Zwecke in Ihrem Anwenderprogramm verwendet werden. Wenn Ihr Anwenderprogramm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.

Die Standardadresse ist AW1000 für PWM1, und AW1002 für PWM2. Der Wert an dieser Adresse bestimmt die Dauer des Impulses und wird jedes Mal auf den oben angegebenen Wert für die "Anfangsimpulsdauer" initialisiert, wenn die CPU vom Betriebszustand STOP in den Betriebszustand RUN wechselt. Sie ändern diesen Ausgangswortwert während der Laufzeit, um die Impulsdauer zu verändern.

## 8.7 Datenprotokollierung

Ihr Steuerungsprogramm kann mit den Anweisungen Data log Laufzeitdatenwerte in beständigen Protokolldateien speichern. Die Datenprotokolldateien werden im Flash-Speicher (CPU oder Memory Card) gespeichert. Die Daten der Protokolldateien werden im herkömmlichen CSV-Format (durch Komma getrennte Werte) gespeichert. Die Datensätze sind in einer zirkulären Protokolldatei vordefinierter Größe organisiert.

Die Anweisungen Data log dienen in Ihrem Programm dazu, einen Datensatz anzulegen, zu öffnen, zu schreiben und die Protokolldateien zu schließen. Sie entscheiden, welche Programmwerte protokolliert werden, indem Sie einen Datenpuffer anlegen, der einen einzigen Protokolldatensatz definiert. Ihr Datenpuffer wird als temporärer Speicher für einen neuen Protokolldatensatz verwendet. Neue aktuelle Werte müssen während der Laufzeit programmatisch in den Puffer übertragen werden. Wenn alle aktuellen Datenwerte aktualisiert sind, können Sie die Anweisung DataLogWrite ausführen, um Daten aus dem Puffer in einen Protokolldatensatz zu übertragen.

Ihre Datenprotokolldateien verwalten Sie mit dem integrierten PLC-Webserver. Laden Sie die letzten Datensätze oder alle Daten herunter, setzen Sie Datensätze zurück oder löschen Sie Protokolldateien. Alle diese Funktionen stehen auf der Standard-Webseite "Datenprotokolle" zur Verfügung. Nachdem Sie eine Datenprotokolldatei auf Ihren PC übertragen haben, können Sie die Daten mit gängigen Tabellenkalkulationsprogrammen wie Excel auswerten.

### 8.7.1 Datensatzstruktur der Datenprotokolle

Die Parameter DATA und HEADER der Anweisung DataLogCreate weisen den Datentyp und die Spaltenkopfbeschreibung aller Datenelemente in einem Protokolldatensatz zu.

#### Parameter DATA für die Anweisung DataLogCreate

Der Parameter DATA verweist auf Speicher, der als temporärer Puffer für einen neuen Protokolldatensatz verwendet wird. Ihm muss eine M- oder DB-Adresse zugewiesen sein.

Zuweisen können Sie einen kompletten DB (abgeleitet von einem PLC-Datentyp, den Sie bei der Erstellung des DBs angeben) oder einen Teil eines DBs (bei dem angegebenen DB-Element kann es sich um einen beliebigen Datentyp, eine Datentypstruktur, einen PLC-Datentyp oder ein Daten-Array handeln).

Die Strukturdatentypen sind auf eine einzige Schachtelungsebene begrenzt. Die Gesamtzahl der deklarierten Datenelemente muss der Anzahl der im Parameter HEADER angegebenen Spalten entsprechen. Sie können maximal 253 Datenelemente (mit Zeitstempel) bzw. 255 Datenelemente (ohne Zeitstempel) zuweisen. Durch diese Einschränkung bleibt Ihr Datensatz innerhalb des Grenzwerts von maximal 256 Spalten eines Excel-Arbeitsblatts.

Der Parameter DATA kann entweder remanente oder nicht remanente Datenelemente in einem DB vom Typ "Standard" (kompatibel mit S7-300/400) oder "Optimiert" zuweisen.

Um einen Protokolldatensatz zu schreiben, müssen Sie zunächst neue Prozesswerte in den temporären Datensatz DATA laden und dann die Anweisung DataLogWrite ausführen, wodurch neue Datensatzwerte in der Datenprotokolldatei gespeichert werden.

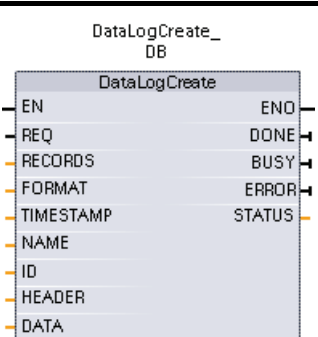
#### Parameter HEADER für die Anweisung DataLogCreate

Der Parameter HEADER zeigt auf die Spaltenköpfe in der obersten Zeile der Datenmatrix in der CSV-Datei. HEADER-Daten müssen sich im DB- oder M-Speicher befinden und die Zeichen müssen den üblichen Formatregeln für CSV-Dateien entsprechen, wobei die einzelnen Spaltennamen durch Komma zu trennen sind. Bei dem Datentyp kann es sich um Strings, Byte-Arrays oder Zeichen-Arrays handeln. Zeichen- bzw. Byte-Arrays ermöglichen eine größere Größe, während der Datentyp String auf maximal 255 Bytes begrenzt ist. Der Parameter HEADER ist optional. Wenn der HEADER nicht zugewiesen ist, wird keine Kopfzeile in der Datenprotokolldatei angelegt.

## 8.7.2 Programmanweisungen zum Steuern von Datenprotokollen

### 8.7.2.1 DataLogCreate

Tabelle 8- 123 Anweisung DataLogCreate

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogCreate_DB" (     req:=_bool_in_,     records:=_udint_in_,     format:=_uint_in_,     timestamp:=_uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     name:=_string_inout_,     ID:=_dword_inout_,     header:=_variant_inout_,     data:=_variant_inout_);</pre>	<p>Erstellt und initialisiert eine Datenprotokolldatei. Die Datei wird im PLC-Verzeichnis \Datenprotokolle angelegt, trägt den im Parameter NAME angegebenen Namen und wird implizit für Schreibanweisungen geöffnet. Mit den Anweisungen Data log können Sie Laufzeit-Prozessdaten programmatisch im Flash-Speicher der CPU speichern.</p> <p>STEP 7 erstellt automatisch den zugehörigen Instanz-DB, wenn Sie die Anweisung einfügen.</p>

<sup>1</sup> Im SCL-Beispiel ist "DataLogCreate\_DB" der Name des Instanz-DBs.

Tabelle 8- 124 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
RECORDS	IN	UDint
FORMAT	IN	UInt
TIMESTAMP	IN	UInt
NAME	IN	Variant

Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)

Die maximale Anzahl Datensätze, die das kreisförmige Datenprotokoll enthalten kann, bevor der älteste Eintrag überschrieben wird:  
Der Datensatz mit der Kopfzeile ist in dieser Anzahl nicht enthalten. Es muss ausreichend PLC-Ladespeicher verfügbar sein, damit das Datenprotokoll erfolgreich angelegt wird. (Voreinstellung - 1)

Format des Datenprotokolls:

- 0 - Internes Format (nicht unterstützt)
- 1 - Durch Komma getrennte Werte, "csv-eng" (Standardwert)

Format des Zeitstempels der Daten: Spaltenköpfe für Datum- und Uhrzeitfelder sind nicht erforderlich. Der Zeitstempel zeigt die Systemzeit (Coordinated Universal Time - UTC) und nicht die Lokalzeit.

- 0 - Kein Zeitstempel
- 1 - Datum und Zeitstempel (Standardwert)

Name des Datenprotokolls: Sie geben einen Namen ein. Diese Variante unterstützt nur den Datentyp String und kann nur im lokalen Speicher, im DB- oder M-Speicher abgelegt werden. (Standardwert: '')

Diese Zeichenkette wird auch für den Namen der Datenprotokolldatei verwendet. Die im Namen enthaltenen Zeichen müssen sich an die Namenseinschränkungen des Windows-Dateisystems halten. Die Zeichen \ / : \* ? " < > | und das Leerzeichen sind nicht zulässig.

Parameter und Datentyp		Datentyp	Beschreibung
ID	Durchgang	DWord	Numerische Kennung des Datenprotokolls: Sie speichern diesen generierten Wert zur Verwendung mit anderen Datenprotokollanweisungen. Der Parameter ID wird bei der Anweisung DataLogCreate nur als Ausgang verwendet. (Standardwert: 0) Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
HEADER	Durchgang	Variant	Pointer auf die Spaltenköpfe des Datenprotokolls in der obersten Zeile der Datenmatrix in der CSV-Datei. (Standardwert: null). HEADER-Daten müssen sich im DB- oder M-Speicher befinden. Die Zeichen müssen den üblichen Formatregeln für CSV-Dateien entsprechen, und die einzelnen Spaltennamen sind durch Komma zu trennen. Bei dem Datentyp kann es sich um Strings, Byte-Arrays oder Zeichen-Arrays handeln. Zeichen- bzw. Byte-Arrays ermöglichen eine größere Größe, während der Datentyp String auf maximal 255 Bytes begrenzt ist. Der Parameter HEADER ist optional. Wenn der HEADER nicht parametrier ist, wird keine Kopfzeile in der Datenprotokolldatei angelegt.
DATA	Durchgang	Variant	Pointer auf die Datenstruktur des Datensatzes, anwenderdefinierter Datentyp (UDT) oder Array. Datensatzdaten müssen sich im DB- oder M-Speicher befinden. Der Parameter DATA gibt die einzelnen Datenelemente (Spalten) eines Protokoll Datensatzes und deren Datentyp an. Die Strukturdatentypen sind auf eine einzige Schachtelungsebene begrenzt. Die Anzahl der deklarierten Datenelemente muss der Anzahl der im Parameter HEADER angegebenen Spalten entsprechen. Sie können maximal 253 Datenelemente (mit Zeitstempel) bzw. 255 Datenelemente (ohne Zeitstempel) zuweisen. Durch diese Einschränkung bleibt Ihr Datensatz innerhalb des Grenzwerts von maximal 256 Spalten in einem Excel-Arbeitsblatt.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Standardwert: Falsch)
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - Keine Anweisung in Bearbeitung</li> <li>• 1 - Anweisung in Bearbeitung</li> </ul>
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)



Es wird eine Datenprotokolldatei mit einer anhand der Parameter RECORDS und DATA fest vorgegebenen Größe angelegt. Die Datensätze sind als kreisförmige Protokolldatei aufgebaut. Neue Datensätze werden in die Datenprotokolldatei eingefügt, bis die maximale Anzahl Datensätze, die im Parameter RECORDS vorgegeben ist, gespeichert ist. Danach überschreibt der nächste Datensatz, der geschrieben wird, den ältesten Datensatz. Eine weitere Schreibanweisung überschreibt den zweitältesten Datensatz usw.

#### Hinweis

Wenn Sie das Überschreiben von Datensätzen verhindern möchten, können Sie mit der Anweisung DataLogNewFile ein neues Datenprotokoll basierend auf dem aktuellen Datenprotokoll anlegen, nachdem das aktuelle Datenprotokoll die maximale Anzahl von Datensätzen gespeichert hat. Neue Datensätze werden dann in der neuen Datenprotokolldatei gespeichert. Die alte Datenprotokolldatei und die Datensatzdaten bleiben im Flash-Speicher erhalten.

Speicherauslastung:

- Die Datenprotokolle belegen nur Ladespeicher.
- Es gibt keinen festen Grenzwert für die Gesamtzahl der Datenprotokolle. Die Größe aller Datenprotokolle insgesamt ist durch die verfügbare Kapazität des Ladespeichers begrenzt. Maximal acht Protokolle dürfen gleichzeitig geöffnet sein.
- Die maximal mögliche Anzahl für den Parameter RECORDS ist der Grenzwert einer UDint-Zahl (4.294.967.295). Der tatsächliche Grenzwert für den Parameter RECORD richtet sich nach der Größe eines einzelnen Datensatzes, der Größe anderer Datenprotokolle und der verfügbaren Kapazität des Ladespeichers. Zudem ist in Excel die Anzahl der Zeilen in einem Excel-Arbeitsblatt begrenzt.

#### Hinweis

Eine Operation DataLogCreate erstreckt sich über viele Programmzyklen. Die tatsächliche für die Erstellung der Protokolldatei benötigte Zeit hängt von der Datensatzstruktur und der Anzahl der Datensätze ab. Ihre Programmlogik muss den Wechsel des DONE-Bits der Anweisung DataLogCreate in den Zustand WAHR überwachen und erfassen, damit das neue Datenprotokoll für andere Datenprotokolloperationen verwendet werden kann.

Tabelle 8- 125 Werte von ERROR und STATUS

ERROR	STATUS (W#16#...)	Beschreibung
0	0000	Kein Fehler
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N <sup>ter</sup> Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.
1	807F	Interner Fehler
1	8090	Ungültiger Dateiname
1	8091	Der Namensparameter ist keine Zeichenkette.

ERROR	STATUS (W#16#...)	Beschreibung
1	8093	Das Datenprotokoll ist bereits vorhanden.
1	8097	Gewünschte Dateilänge überschreitet maximale Dateigröße für das Dateisystem.
1	80B3	Nicht genügend Ladespeicher.
1	80B4	MC (Speichermodul) ist schreibgeschützt.
1	80C1	Zu viele geöffnete Dateien: Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein.
1	8253	Ungültige Anzahl Datensätze
1	8353	Ungültige Formatauswahl
1	8453	Ungültige Zeitstempelauswahl
1	8B24	Ungültige HEADER-Bereichszuweisung: Beispiel: Verweist auf lokalen Speicher
1	8B51	Ungültiger Datentyp für Parameter HEADER
1	8B52	Zu viele Datenelemente im Parameter HEADER
1	8C24	Ungültige DATA-Bereichszuweisung: Beispiel: Zeigt auf lokalen Speicher
1	8C51	Ungültiger Datentyp für Parameter DATA
1	8C52	Zu viele Datenelemente im Parameter DATA

### 8.7.2.2 DataLogOpen

Tabelle 8- 126 Anweisung DataLogOpen

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogOpen_DB" (   req:=_bool_in_,   mode:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   name:=_string_inout_,   ID:=_dword_inout_);</pre>	<p>Öffnet eine bereits vorhandene Datenprotokolldatei. Ein Datenprotokoll muss geöffnet sein, damit Sie neue Datensätze in das Protokoll schreiben können. Datenprotokolle lassen sich einzeln öffnen und schließen. Maximal acht Datenprotokolle können gleichzeitig geöffnet sein.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

<sup>2</sup> Im SCL-Beispiel ist "DataLogOpen\_DB" der Name des Instanz-DBs.

Tabelle 8- 127 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)
MODE	IN	UInt Betriebsart: <ul style="list-style-type: none"> <li>0 - An vorhandene Daten anhängen (Standardwert)</li> <li>1 - Alle vorhandenen Datensätze löschen</li> </ul>
NAME	IN	Variant Name eines vorhandenen Datenprotokolls: Diese Variante unterstützt nur den Datentyp String und kann nur im lokalen Speicher, im DB- oder M-Speicher abgelegt werden. (Standardwert: '')

Parameter und Datentyp		Datentyp	Beschreibung
ID	Durchgang	DWord	Numerische Kennung eines Datenprotokolls. (Standardwert: 0) <b>Hinweis:</b> Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Standardwert: Falsch)
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - Keine Anweisung in Bearbeitung</li> <li>• 1 - Anweisung in Bearbeitung</li> </ul>
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Sie können entweder den NAMEN oder eine ID (Parameter ID als Eingang) eines bereits vorhandenen Datenprotokolls eingeben. Wenn Sie beide Parameter angeben, und eine gültige ID entspricht dem Datenprotokoll NAME, dann wird die ID verwendet und der NAME wird ignoriert.

Der NAME muss dem Namen eines mit der Anweisung DataLogCreate angelegten Datenprotokolls entsprechen. Wird nur der NAME angegeben, und der NAME verweist auf ein gültiges Datenprotokoll, dann wird die entsprechende ID ausgegeben (Parameter ID als Ausgang).

#### Hinweis

##### Allgemeine Verwendung von Datenprotokolldateien

- Nach Ausführung der Anweisungen DataLogCreate und DataLogNewFile werden automatisch Datenprotokolldateien geöffnet.
- Nach einem Wechsel des PLC-Geräts von RUN in STOP oder nach einem Neustart des PLC-Geräts werden Datenprotokolldateien automatisch geschlossen.
- Damit eine neue Anweisung DataLogWrite durchgeführt werden kann, muss eine Datenprotokolldatei geöffnet sein.
- Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein. Mehr als acht Datenprotokolldateien dürfen vorhanden sein, von diesen müssen jedoch einige geschlossen werden, so dass maximal acht gleichzeitig geöffnet sind.

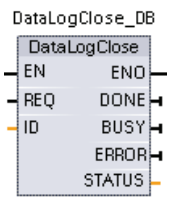
Tabelle 8- 128 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	0002	Warnung: Datenprotokolldatei wurde von diesem Anwendungsprogramm bereits geöffnet
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N <sup>ter</sup> Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.

ERROR	STATUS (W#16#)	Beschreibung
1	8090	Datenprotokolldefinition stimmt nicht mit der vorhandenen Datenprotokolldatei überein.
1	8091	Der Namensparameter ist keine Zeichenkette.
1	8092	Datenprotokoll ist nicht vorhanden.
1	80C0	Datenprotokolldatei ist gesperrt.
1	80C1	Zu viele geöffnete Dateien: Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein.

### 8.7.2.3 DataLogClose

Tabelle 8- 129 Anweisung DataLogClose

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogClose_DB" (     req:=_bool_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     ID:=_dword_inout_);</pre>	<p>Schließt eine geöffnete Datenprotokolldatei. Eine für ein geschlossenes Datenprotokoll ausgeführte Anweisung DataLogWrite führt zu einem Fehler. Schreibenweisungen für dieses Datenprotokoll sind erst zulässig, nachdem eine Anweisung DataLogOpen ausgeführt wurde.</p> <p>Beim Wechsel in den Betriebszustand STOP werden alle geöffneten Datenprotokolldateien geschlossen.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

<sup>2</sup> Im SCL-Beispiel ist "DataLogClose\_DB" der Name des Instanz-DBs.

Tabelle 8- 130 Datentypen für die Parameter

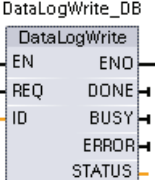
Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)
ID	Durchgang	DWord Numerische Kennung eines Datenprotokolls. Wird bei der Anweisung DataLogClose nur als Eingang verwendet. (Standardwert: 0) <b>Hinweis:</b> Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool <ul style="list-style-type: none"> <li>0 - Keine Anweisung in Bearbeitung</li> <li>1- Anweisung in Bearbeitung</li> </ul>
ERROR	OUT	Bool Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word Ausführungsbedingung (Standardwert: 0)

Tabelle 8- 131 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	0001	Datenprotokoll nicht geöffnet
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N <sup>ter</sup> Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8092	Datenprotokoll ist nicht vorhanden.

### 8.7.2.4 DataLogWrite

Tabelle 8- 132 Anweisung DataLogWrite

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogWrite_DB" (     req:=_bool_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     ID:=_dword_inout_);</pre>	<p>Schreibt einen Datensatz in das angegebene Datenprotokoll. Das bereits vorhandene Zieldatenprotokoll muss geöffnet sein, damit eine Anweisung DataLogWrite durchgeführt werden kann.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

<sup>2</sup> Im SCL-Beispiel ist "DataLogWrite\_DB" der Name des Instanz-DBs.

Tabelle 8- 133 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)
ID	Durchgang	DWord Numerische Kennung des Datenprotokolls: Wird bei der Anweisung DataLogWrite nur als Eingang verwendet. (Standardwert: 0) <b>Hinweis:</b> Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool <ul style="list-style-type: none"> <li>• 0 - Keine Anweisung in Bearbeitung</li> <li>• 1 - Anweisung in Bearbeitung</li> </ul>
ERROR	OUT	Bool Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word Ausführungsbedingung (Standardwert: 0)

Die Speicheradresse und die Datenstruktur des Datensatzpuffers werden vom Parameter DATA der Anweisung DataLogCreate konfiguriert. Um neue Datensatzdaten aus dem Puffer in ein Datenprotokoll zu übertragen, müssen Sie den Datensatzpuffer programmatisch mit aktuellen Laufzeitprozesswerten laden und dann die Anweisung DataLogWrite ausführen.

Der Parameter ID gibt eine Datenprotokoll- und Datensatzkonfiguration an. Die ID-Nummer wird beim Anlegen eines Datenprotokolls generiert.

Enthält die kreisförmige Datenprotokolldatei leere Datensätze, wird in den nächsten freien Datensatz geschrieben. Sind alle Datensätze belegt, wird der älteste Datensatz überschrieben.

<b>VORSICHT</b>
<b>Möglicher Protokolldatenverlust während eines CPU-Spannungsausfalls</b>
Tritt während einer noch nicht beendeten Anweisung DataLogWrite ein Spannungsausfall auf, geht der ins Datenprotokoll übertragene Datensatz möglicherweise verloren.

Tabelle 8- 134 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	0001	Zeigt an, dass das Datenprotokoll voll ist: Jedes Datenprotokoll wird mit einer vorgegebenen Maximalanzahl von Datensätzen angelegt. Der letzte Datensatz dieser Maximalanzahl wurde geschrieben. Die nächste Schreibenweisung überschreibt den ältesten Datensatz.
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N <sup>ter</sup> Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.
1	8092	Datenprotokoll ist nicht vorhanden.
1	80B0	Datenprotokolldatei ist nicht geöffnet (nur beim Modus mit explizitem Öffnen).

### 8.7.2.5 DataLogNewFile

Tabelle 8- 135 Anweisung DataLogNewFile

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogNewFile_DB" (   req:=_bool_in_,   records:=_udint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   name:=_DataLog_out_,   ID:=_dword_inout_);</pre>	<p>Ermöglicht Ihrem Programm, eine neue Datenprotokolldatei basierend auf einer bestehenden Datenprotokolldatei anzulegen.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

<sup>2</sup> Im SCL-Beispiel ist "DataLogNewFile\_DB" der Name des Instanz-DBs.

Tabelle 8- 136 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)
RECORDS	IN	UDInt Die maximale Anzahl Datensätze, die das kreisförmige Datenprotokoll enthalten kann, bevor der älteste Eintrag überschrieben wird. (Standardwert: 1) Der Datensatz mit der Kopfzeile ist in dieser Anzahl nicht enthalten. Es muss ausreichend CPU-Ladespeicher verfügbar sein, damit das Datenprotokoll erfolgreich angelegt wird.
NAME	IN	Variant Name des Datenprotokolls: Sie geben einen Namen ein. Diese Variante unterstützt nur den Datentyp String und kann nur im lokalen Speicher, im DB- oder M-Speicher abgelegt werden. (Standardwert: ' ') Diese Zeichenkette wird auch für den Namen der Datenprotokolldatei verwendet. Die im Namen enthaltenen Zeichen müssen sich an die Namensbeschränkungen des Windows-Dateisystems halten. Die Zeichen \ / : * ? " < >   und das Leerzeichen sind nicht zulässig.)
ID	Durchgang	DWord Numerische Kennung des Datenprotokolls (Standardwert: 0): <ul style="list-style-type: none"> <li>Während der Ausführung gibt der Eingang ID ein gültiges Datenprotokoll an. Die neue Datenprotokollkonfiguration wird aus diesem Datenprotokoll kopiert.</li> <li>Nach der Ausführung wird der Parameter ID zu einem Ausgang, der die ID der neu angelegten Datenprotokolldatei ausgibt.</li> </ul> <b>Hinweis:</b> Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool <ul style="list-style-type: none"> <li>0 - Keine Anweisung in Bearbeitung</li> <li>1 - Anweisung in Bearbeitung</li> </ul>
ERROR	OUT	Bool Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word Ausführungsbedingung (Standardwert: 0)

Sie können die Anweisung DataLogNewFile ausführen, wenn ein Datenprotokoll voll ist oder nahezu voll ist und Sie keine Daten verlieren möchten, die im Datenprotokoll gespeichert sind. Basierend auf der Struktur der vollen Datenprotokolldatei kann eine neue, leere Datenprotokolldatei angelegt werden. Der Kopfdatensatz wird mit den Eigenschaften des ursprünglichen Datenprotokolls (DATA-Einstellungen für Datensatzpuffer, Datenformat und Zeitstempel) aus dem ursprünglichen Datenprotokoll dupliziert. Die ursprüngliche Datenprotokolldatei wird implizit geschlossen und die neue Datenprotokolldatei implizit geöffnet.

Auslösung von DataLogWrite durch Parameter: Ihr Programm muss die Parameter ERROR und STATUS jeder Anweisung DataLogWrite überwachen. Wenn der letzte Datensatz geschrieben wurde und ein Datenprotokoll voll ist, lautet das ERROR-Bit der Anweisung DataLogWrite = 1 und das Datenwort STATUS der Anweisung DataLogWrite = 1. Diese ERROR- und STATUS-Werte sind nur einen Zyklus lang gültig, deshalb muss Ihre Überwachungslogik ERROR = 1 als Zeitfenster nutzen, um den STATUS-Wert zu erfassen und dann auf STATUS = 1 zu prüfen (Datenprotokoll ist voll).

Funktionsweise von DataLogNewFile: Wenn Ihre Programmlogik das Signal "Datenprotokoll voll" erhält, wird mit diesem Zustand eine Anweisung DataLogNewFile aktiviert. Sie müssen Anweisung DataLogNewFile mit der ID eines vorhandenen (üblicherweise vollen) und geöffneten Datenprotokolls ausführen, der Parameter NAME muss jedoch neu und eindeutig sein. Nach Ausführung der Anweisung DataLogNewFile wird ein neuer ID-Wert für das Datenprotokoll ausgegeben (als Ausgangsparameter), der dem Namen des neuen Datenprotokolls entspricht. Die neue Datenprotokolldatei wird implizit geöffnet und kann neue Datensätze speichern. Neue Anweisungen DataLogWrite, die sich an die neue Datenprotokolldatei richten, müssen den von der Anweisung DataLogNewFile ausgegebenen ID-Wert nutzen.

**Hinweis**

Eine Operation DataLogNewFile erstreckt sich über viele Programmzyklen. Die tatsächliche für die Erstellung der Protokolldatei benötigte Zeit hängt von der Datensatzstruktur und der Anzahl der Datensätze ab. Ihre Programmlogik muss den Wechsel des DONE-Bits der Anweisung DataLogNewFile in den Zustand WAHR überwachen und erfassen, damit das neue Datenprotokoll für andere Datenprotokolloperationen verwendet werden kann.

Tabelle 8- 137 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N <sup>ter</sup> Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.
1	8090	Ungültiger Dateiname
1	8091	Der Namensparameter ist keine Zeichenkette.
1	8092	Datenprotokoll ist nicht vorhanden.
1	8093	Das Datenprotokoll ist bereits vorhanden.
1	8097	Gewünschte Dateilänge überschreitet maximale Dateigröße für das Dateisystem.
1	80B3	Nicht genügend Ladespeicher.
1	80B4	MC ist schreibgeschützt.
1	80C1	Zu viele geöffnete Dateien.

**8.7.3 Arbeiten mit Datenprotokollen**

Die Datenprotokolldateien werden im beständigen Flash-Speicher im CSV-Format (durch Komma getrennte Werte) gespeichert. Sie können die Datenprotokolle über den PLC-Webserver anzeigen oder indem Sie die Memory Card des PLC-Geräts entnehmen und in einen herkömmlichen PC-Kartenleser einlegen.



## Datenprotokolle über den PLC-Webserver anzeigen

Wenn der PROFINET-Anschluss des PLC-Geräts und ein PC an ein Netzwerk angeschlossen sind, können Sie einen PC-Webbrowser wie Microsoft Internet Explorer oder Mozilla Firefox nutzen, um auf den integrierten PLC-Webserver zuzugreifen. Das PLC-Gerät kann sich, während Sie den PLC-Webserver bedienen, in der Betriebsart RUN oder STOP befinden. Befindet sich das PLC-Gerät im Betriebszustand RUN, wird Ihr Steuerungsprogramm weiterhin ausgeführt, während der PLC-Webserver über das Netzwerk Protokoll Daten überträgt.

Zugriff auf den Webserver:

1. Aktivieren Sie den Webserver in der Gerätekonfiguration der Ziel-CPU (Seite 532).
2. Verbinden Sie Ihren PC über das PROFINET-Netzwerk mit dem Zielsystem (Seite 533).
3. Melden Sie sich am integrierten Webserver an (Seite 535).
4. Laden Sie die letzten Datensätze oder alle Daten herunter, setzen Sie Datensätze zurück oder löschen Sie Protokolldateien. Alle diese Funktionen stehen auf der Standard-Webseite "Datenprotokolle" zur Verfügung (Seite 546).
5. Wenn Sie eine Kopie der Datenprotokolldatei auf Ihren PC herunterladen, können Sie die CSV-Datei mit einem Tabellenkalkulationsprogramm wie Excel öffnen.

## Datenprotokolle auf einer Memory Card des PLC-Geräts anzeigen

Ist in der S7-1200 CPU eine Memory Card vom Typ "Programmkarte" gesteckt, können Sie die Memory Card entnehmen und in einen Standardkartensteckplatz wie SD (Secure Digital) oder MMC (MultiMediaCard) an einem PC oder PG einfügen. Das PLC-Gerät ist beim Entnehmen der Memory Card im Betriebszustand STOP und Ihr Steuerungsprogramm wird nicht ausgeführt.

Öffnen Sie dann den Windows Explorer für Dateien und navigieren Sie zum Verzeichnis \Datenprotokoll auf der Memory Card. In diesem Verzeichnis befinden sich alle Ihre CSV-Datenprotokolldateien.

Kopieren Sie die Datenprotokolldateien in ein lokales Laufwerk auf Ihrem PC. Dann können Sie eine lokale Kopie einer CSV-Datei mit Excel öffnen; dies ist nicht die ursprüngliche Datei, die auf der Memory Card gespeichert ist.

### VORSICHT

**Sie können Datenprotokolldateien über einen PC-Kartenleser von einer S7-1200 Memory Card kopieren, jedoch noch ändern oder löschen.**

Es ist empfehlenswert, die Standardseite des Webserverdatenprotokolls zu nutzen, um Datenprotokolldateien anzuzeigen, herunterzuladen (zu kopieren), Daten darin zu löschen und komplette Datenprotokolldateien zu löschen. Der Webserver verwaltet die Dateien auf der Memory Card und verhindert das versehentliche Ändern oder Löschen von Daten.

Das direkte Aufrufen des Dateisystems der Memory Card im Windows Explorer hat den Nachteil, dass Sie versehentlich Datenprotokoll- oder andere Systemdateien löschen oder ändern können, so dass möglicherweise eine Datei beschädigt oder die Memory Card unbrauchbar wird.

### Datenprotokolle in einem Webbrowser anzeigen

Auch wenn Sie die Webserverfunktion nicht nutzen, können Sie Datenprotokolle direkt in einem Webbrowser wie Internet Explorer oder Mozilla Firefox anzeigen. Geben Sie dafür einfach den folgenden Text in die Adressleiste Ihres Browsers ein. Verwenden Sie dabei statt "MyDataLog" die IP-Adresse Ihrer CPU und den Namen der Datenprotokolldatei, den Sie in STEP 7 angegeben haben:

```
http://192.168.0.1/DataLog.html?FileName=MyDataLog.csv
```

Die festen Adressen von Datenprotokolldateien ermöglichen es zudem, über Dateierfassungswerkzeuge anderer Hersteller auf diese Dateien zuzugreifen.

### 8.7.4 Grenzwerte für die Größe von Datenprotokolldateien

Datenprotokolldateien nutzen den Ladespeicher des PLC-Geräts gemeinsam mit dem Programm, mit Programmdateien, Konfigurationsdateien, benutzerdefinierten Webseiten und PLC-Systemdateien. Ein großes Programm, das den internen Ladespeicher nutzt, belegt viel Platz im Ladespeicher. Möglicherweise steht dadurch nicht genügend freier Speicherplatz für Datenprotokolldateien zur Verfügung. In diesem Fall können Sie eine Memory Card als "Programmkarte" nutzen, um die Kapazität des Ladespeichers zu erhöhen. S7-1200 CPUs können entweder internen oder externen Ladespeicher nutzen, jedoch nicht beide gleichzeitig.

Im Kapitel zur Memory Card finden Sie Einzelheiten zum Anlegen einer Programmkarte (Seite 121).

### Regel für die maximale Größe einer Datenprotokolldatei

Die maximale Größe einer Datenprotokolldatei darf 25 % der Größe des Ladespeichers (intern oder extern) nicht überschreiten. Sollte Ihre Anwendung weitere Datenprotokolleinträge benötigen, erstellen Sie mit der Anweisung "DataLogNewFile" eine neue Datei, wenn alle Datensätze in der ersten Datei belegt sind. In der folgenden Tabelle finden Sie die Angaben für die maximale Größe einer Datenprotokolldatei.

Tabelle 8- 138 Größe des Ladespeichers und maximale Größe für eine Datenprotokolldatei

Datenbereich	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	Datenspeicherung
<b>Interner Ladespeicher</b> Flash-Speicher	1 MB (max. 250 KB für eine Datenprotokolldatei)	1 MB (max. 250 KB für eine Datenprotokolldatei)	4 MB (max. 500 KB für eine Datenprotokolldatei)	4 MB (max. 500 KB für eine Datenprotokolldatei)	Anwenderprogramm und Programmdateien, Konfigurationsdateien, Datenprotokolle, benutzerdefinierte Webseiten und PLC-Systemdateien
<b>Externer Ladespeicher</b> Optionale Memory Card als "Programmkarte"	2 MB, 12 MB oder 24 MB je nach Größe der SD-Karte (max. 500 KB für eine Datenprotokolldatei bei Verwendung einer 2-MB-Karte) (max. 6 MB für eine Datenprotokolldatei bei Verwendung einer 24-MB-Karte)				

### Größe des freien Ladespeichers ermitteln

1. Stellen Sie eine Online-Verbindung zwischen STEP 7 und dem S7-1200 Zielsystem her.
2. Laden Sie das Programm, in das Sie Datenprotokollanweisungen einfügen möchten.
3. Erstellen Sie optional anwenderdefinierte Webseiten. (Die Standard-Webseiten für den Zugriff auf Datenprotokolle sind in der Firmware des PLC-Geräts gespeichert und belegen keinen Platz im Ladespeicher.)
4. Fragen Sie mit den Online- und Diagnosefunktionen die Größe des Ladespeichers und den Prozentwert des freien Ladespeichers ab (Seite 723).
5. Multiplizieren Sie die Größe des Ladespeichers mit dem Prozentwert des freien Speicherplatzes, um die gegenwärtige Kapazität des freien Ladespeichers zu erhalten.

### Regel für die maximale Größe für alle Datenprotokolle insgesamt

Die Kapazität des freien Ladespeichers variiert im Normalbetrieb, weil das Betriebssystem ständig Speicher benötigt und wieder freigibt. Sie sollten die Gesamtgröße für alle Datenprotokolldateien auf die Hälfte des verfügbaren freien Speicherplatzes begrenzen.

### Speicherbedarf für einen Datensatz eines Datenprotokolls berechnen

Die Daten in Datenprotokolldateien werden als Zeichenbytes im CSV-Format (durch Komma getrennte Werte) gespeichert. Die folgende Tabelle zeigt die Anzahl Bytes, die erforderlich sind, um jeden Datentyp zu speichern.

Tabelle 8- 139 Datengrößen von CSV-Dateien

Datentyp	Anzahl der Bytes (Datenbytes plus Byte für Komma-Trennzeichen)
Bool	2
Byte	5
Wort	7
DWord	12
Char	4
String	<p><b>Beispiel 1:</b> MeinString[10] Die maximale Zeichenkettengröße wird mit 10 Zeichen angegeben. Textzeichen + automatische Auffüllung mit Leerzeichen = 10 Byte Anführungszeichen am Anfang und am Ende + Kommazeichen = 3 Byte 10 + 3 = 13 Byte insgesamt</p> <p><b>Beispiel 2:</b> MeinString2 Wird keine Größe in eckigen Klammern angegeben, dann werden standardmäßig 254 Byte zugewiesen. Textzeichen + automatische Auffüllung mit Leerzeichen = 254 Byte Anführungszeichen am Anfang und am Ende + Kommazeichen = 3 Byte 254 + 3 = 257 Byte insgesamt</p>
USInt	5
UInt	7
UDInt	12

<b>Datentyp</b>	<b>Anzahl der Bytes (Datenbytes plus Byte für Komma-Trennzeichen)</b>
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Time	15
DTL	24

Der Parameter DATA der Anweisung DataLogCreate zeigt auf eine Struktur, die die Anzahl der Datenfelder und den Datentyp jedes Datenfelds für einen Protokolldatensatz angibt. Die Tabelle oben zeigt die erforderliche Anzahl Bytes in der CSV-Datei für jeden Datentyp. Multiplizieren Sie die Anzahl des jeweiligen Datentyps mit der erforderlichen Anzahl Bytes für diesen Datentyp. Führen Sie diese Berechnung für jeden Datentyp in dem Datensatz durch und addieren Sie die Anzahl Bytes, um die Gesamtgröße des Datensatzes zu ermitteln. Addieren Sie ein Byte für das Zeilenendezeichen.

Größe des Protokolldatensatzes = Addition der erforderlichen Bytes für alle Datenfelder + 1 (Zeilenendezeichen).

### **Speicherbedarf für eine gesamte Datenprotokolldatei berechnen**

Der Parameter RECORDS der Anweisung DataLogCreate legt die maximale Anzahl Datensätze in einer Datenprotokolldatei fest. Beim Anlegen der Datenprotokolldatei wird die maximale Speichergröße zugeordnet.

Größe der Datenprotokolldatei = (Anzahl der Bytes in einem Datensatz) x (Anzahl der Datensätze).

## 8.7.5 Beispielprogramm für Datenprotokolle

Dieses Beispielprogramm für Datenprotokolle zeigt nicht die gesamte Programmlogik, die erforderlich ist, um Abtastwerte eines dynamischen Prozesses abzurufen, es zeigt jedoch die wesentliche Funktionsweise der Anweisungen für Datenprotokolle. Die Struktur und Anzahl der verwendeten Protokolldateien richtet sich nach Ihren Prozesssteuerungsanforderungen.

### Hinweis

#### Allgemeine Verwendung von Datenprotokolldateien

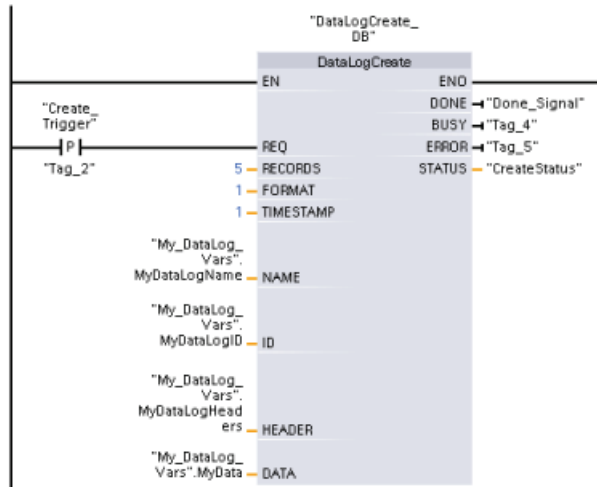
- Nach Ausführung der Anweisungen DataLogCreate und DataLogNew werden automatisch Datenprotokolldateien geöffnet.
- Nach einem Wechsel des PLC-Geräts von RUN in STOP oder nach einem Neustart des PLC-Geräts werden Datenprotokolldateien automatisch geschlossen.
- Damit eine Anweisung DataLogWrite durchgeführt werden kann, muss eine Datenprotokolldatei geöffnet sein.
- Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein. Mehr als acht Datenprotokolldateien dürfen vorhanden sein, von diesen müssen jedoch einige geschlossen werden, so dass maximal acht gleichzeitig geöffnet sind.

## Beispielprogramm für Datenprotokolle

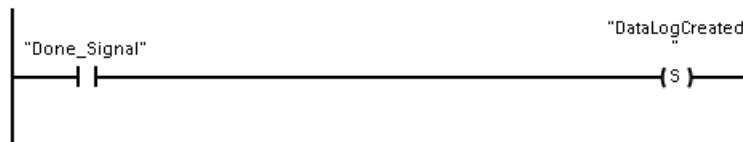
Beispielhafte Datenprotokollnamen, Kopfzeilentext und die Struktur MyData werden in einem Datenbaustein erstellt. Die drei MyData-Variablen speichern neue Abtastwerte temporär. Die Prozessabtastwerte an diesen DB-Adressen werden durch Ausführung der Anweisung DataLogWrite in eine Datenprotokolldatei übertragen.

My_Datalog_Vars			
	Name	Datentyp	Startwert
1	Static		
2	MyNEWDatalogName	String	'MyNEWDatalog'
3	MyDatalogName	String	'MyDatalog'
4	MyDatalogID	DWord	0
5	MyDatalogHeaders	String	'Count,Temperature,Pressure'
6	MyData	Struct	
7	MyCount	Int	0
8	MyTemperature	Real	0.0
9	MyPressure	Real	0.0

**Netzwerk 1** Eine steigende Flanke an REQ startet die Datenprotokollerstellung.



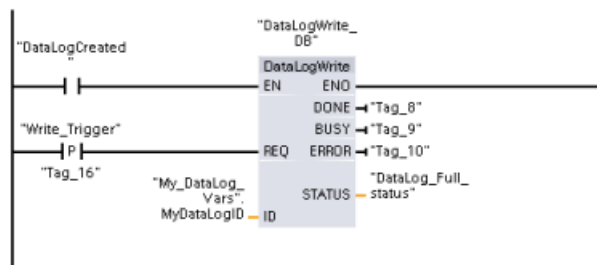
**Netzwerk 2** Erfassen Sie den Ausgang DONE von DataLogCreate, weil er nur einen Zyklus lang gültig ist.



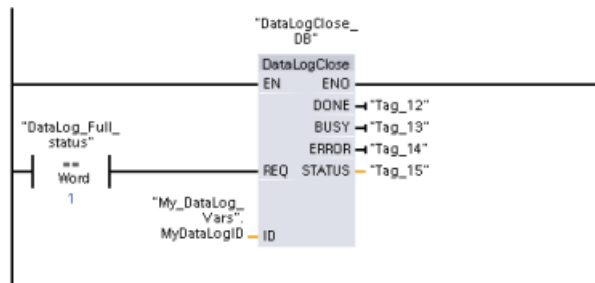
**Netzwerk 3** Eine positive Flanke löst den Zeitpunkt aus, an dem neue Prozesswerte in der Struktur MyData gespeichert werden.



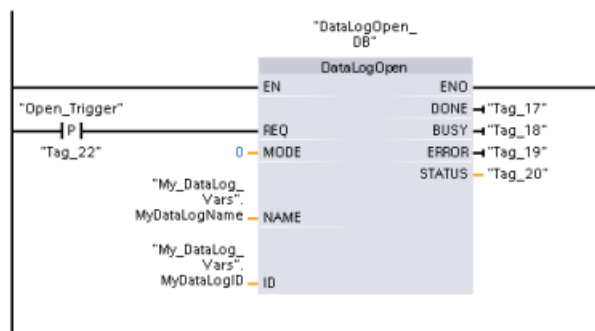
**Netzwerk 4** Der Zustand des Eingangs EN basiert auf dem Zeitpunkt, zu dem die Ausführung von DataLogCreate beendet ist. Ein Erstellungsvorgang erstreckt sich über viele Zyklen und muss beendet sein, damit ein Schreibvorgang durchgeführt werden kann. Die positive Flanke am Eingang REQ ist das Ereignis, das eine aktivierte Schreiboperation auslöst.



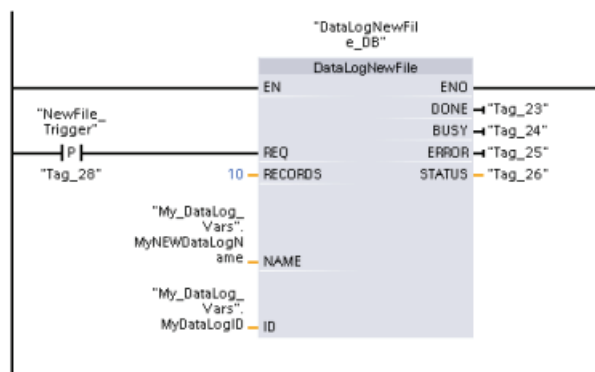
**Netzwerk 5** Schließen Sie das Datenprotokoll, nachdem der letzte Datensatz geschrieben wurde. Nach Ausführung der Anweisung DataLogWrite, die den letzten Datensatz schreibt, wird der Zustand "Protokolldatei voll" gemeldet, wenn der Ausgang STATUS von DataLogWrite = 1 ist.



**Netzwerk 6** Eine positive Flanke am Eingang REQ der Anweisung DataLogOpen simuliert, dass der Anwender an einem HMI-Gerät eine Taste drückt, die eine Datenprotokolldatei öffnet. Wenn Sie eine Datenprotokolldatei öffnen, in der alle Datensätze mit Prozessdaten belegt sind, dann überschreibt die nächste Ausführung der Anweisung DataLogWrite den ältesten Datensatz. Sie können jedoch auch das alte Datenprotokoll aufbewahren und stattdessen ein neues Datenprotokoll anlegen. Dies wird in Netzwerk 7 gezeigt.



**Netzwerk 7** Der ID-Parameter ist ein IN/OUT-Typ. Zunächst geben Sie den ID-Wert des vorhandenen Datenprotokolls an, dessen Struktur Sie kopieren möchten. Nachdem die Anweisung DataLogNewFile durchgeführt wurde, wird ein neuer und eindeutiger ID-Wert für das neue Datenprotokoll in die Adresse des ID-Verweises zurückgeschrieben. Die erforderliche Erfassung ONE-Bit = WAHR wird nicht gezeigt. Ein Beispiel für die Logik des DONE-Bits finden Sie in den Netzwerken 1, 2 und 4.



Vom Beispielprogramm angelegte und mit dem Webserver der S7-1200 CPU angezeigte Datenprotokolldateien

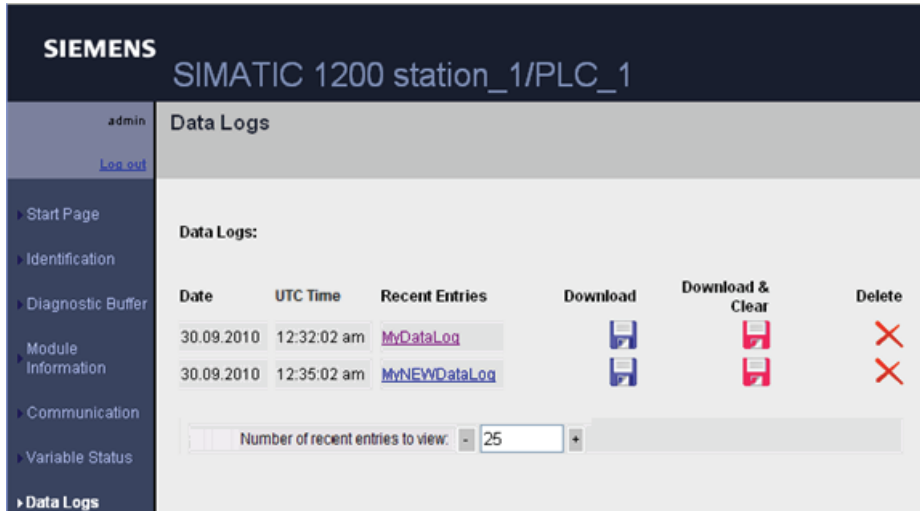


Tabelle 8- 140 In Excel angezeigte, heruntergeladene Beispiele für CSV-Dateien

<p>Zwei geschriebene Datensätze in einer Datei, die maximal fünf Datensätze enthalten kann</p>	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Record</td> <td>Date</td> <td>UTC Time</td> <td>Count</td> <td>Temperature</td> <td>Pressure</td> </tr> <tr> <td>2</td> <td>1</td> <td>9/29/2010</td> <td>21:01:46</td> <td>5</td> <td>5.00E+00</td> <td>5.00E+00</td> </tr> <tr> <td>3</td> <td>2</td> <td>9/29/2010</td> <td>21:01:47</td> <td>5</td> <td>5.00E+00</td> <td>5.00E+00</td> </tr> <tr> <td>4</td> <td colspan="6">//END</td> </tr> <tr> <td>5</td> <td colspan="6"></td> </tr> </tbody> </table>		A	B	C	D	E	F	1	Record	Date	UTC Time	Count	Temperature	Pressure	2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00	3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00	4	//END						5																				
	A	B	C	D	E	F																																																			
1	Record	Date	UTC Time	Count	Temperature	Pressure																																																			
2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00																																																			
3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00																																																			
4	//END																																																								
5																																																									
<p>Fünf Datensätze in einer Datenprotokolldatei, die maximal fünf Datensätze enthalten kann</p>	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Record</td> <td>Date</td> <td>UTC Time</td> <td>Count</td> <td>Temperature</td> <td>Pressure</td> </tr> <tr> <td>2</td> <td>1</td> <td>9/30/2010</td> <td>20:26:56</td> <td>1</td> <td>9.86E+01</td> <td>3.52E+01</td> </tr> <tr> <td>3</td> <td>2</td> <td>9/30/2010</td> <td>20:28:43</td> <td>2</td> <td>1.00E+02</td> <td>3.73E+01</td> </tr> <tr> <td>4</td> <td>3</td> <td>9/30/2010</td> <td>20:29:03</td> <td>3</td> <td>9.99E+01</td> <td>3.68E+01</td> </tr> <tr> <td>5</td> <td>4</td> <td>9/30/2010</td> <td>20:29:21</td> <td>4</td> <td>9.95E+01</td> <td>3.64E+01</td> </tr> <tr> <td>6</td> <td>5</td> <td>9/30/2010</td> <td>20:30:19</td> <td>5</td> <td>9.92E+01</td> <td>3.74E+01</td> </tr> <tr> <td>7</td> <td colspan="6"></td> </tr> </tbody> </table>		A	B	C	D	E	F	1	Record	Date	UTC Time	Count	Temperature	Pressure	2	1	9/30/2010	20:26:56	1	9.86E+01	3.52E+01	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01	7						
	A	B	C	D	E	F																																																			
1	Record	Date	UTC Time	Count	Temperature	Pressure																																																			
2	1	9/30/2010	20:26:56	1	9.86E+01	3.52E+01																																																			
3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01																																																			
4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01																																																			
5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01																																																			
6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01																																																			
7																																																									
<p>Nachdem ein weiterer Datensatz in die oben abgebildete Datei, die bereits voll ist, geschrieben wurde, überschreibt der sechste Schreibvorgang den ältesten Datensatz 1 mit Datensatz 6. Ein weiterer Schreibvorgang überschreibt Datensatz 2 mit Datensatz 7 usw.</p>	<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Record</td> <td>Date</td> <td>UTC Time</td> <td>Count</td> <td>Temperature</td> <td>Pressure</td> </tr> <tr> <td>2</td> <td>6</td> <td>9/30/2010</td> <td>20:32:03</td> <td>6</td> <td>9.86E+01</td> <td>3.58E+01</td> </tr> <tr> <td>3</td> <td>2</td> <td>9/30/2010</td> <td>20:28:43</td> <td>2</td> <td>1.00E+02</td> <td>3.73E+01</td> </tr> <tr> <td>4</td> <td>3</td> <td>9/30/2010</td> <td>20:29:03</td> <td>3</td> <td>9.99E+01</td> <td>3.68E+01</td> </tr> <tr> <td>5</td> <td>4</td> <td>9/30/2010</td> <td>20:29:21</td> <td>4</td> <td>9.95E+01</td> <td>3.64E+01</td> </tr> <tr> <td>6</td> <td>5</td> <td>9/30/2010</td> <td>20:30:19</td> <td>5</td> <td>9.92E+01</td> <td>3.74E+01</td> </tr> <tr> <td>7</td> <td colspan="6"></td> </tr> </tbody> </table>		A	B	C	D	E	F	1	Record	Date	UTC Time	Count	Temperature	Pressure	2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01	7						
	A	B	C	D	E	F																																																			
1	Record	Date	UTC Time	Count	Temperature	Pressure																																																			
2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01																																																			
3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01																																																			
4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01																																																			
5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01																																																			
6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01																																																			
7																																																									



## 8.8 Datenbausteinststeuerung

### 8.8.1 READ\_DBL, WRIT\_DBL (Aus DB im Ladespeicher lesen, In DB im Ladespeicher schreiben)

Tabelle 8- 141 Anweisungen READ\_DBL und WRIT\_DBL

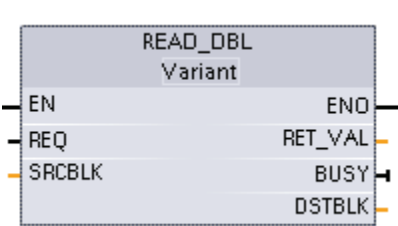
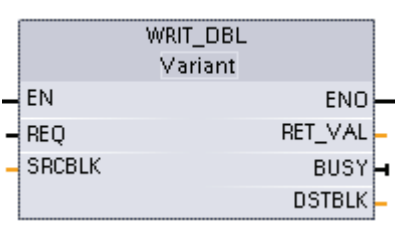
KOP/FUP		Beschreibung
	<pre> READ_DBL (   req:=_bool_in_,   srcblk:=_variant_in_,   busy=&gt;_bool_out_,   dstblk=&gt;_variant_out_); </pre>	<p>Kopiert DB-Startwerte oder einen Teil der Werte aus dem Ladespeicher in einen Ziel-DB im Arbeitsspeicher.</p> <p>Der Inhalt des Ladespeichers wird während des Kopiervorgangs nicht verändert.</p>
	<pre> WRIT_DBL (   req:=_bool_in_,   srcblk:=_variant_in_,   busy=&gt;_bool_out_,   dstblk=&gt;_variant_out_); </pre>	<p>Kopiert aktuelle DB-Werte oder einen Teil der Werte aus dem Arbeitsspeicher in einen Ziel-DB im Ladespeicher.</p> <p>Der Inhalt des Arbeitsspeichers wird während des Kopiervorgangs nicht verändert.</p>

Tabelle 8- 142 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	BOOL
SRCBLK	IN	VARIANT
RET_VAL	OUT	INT
BUSY	OUT	BOOL
DSTBLK	OUT	VARIANT

Typischerweise wird ein DB sowohl im Ladespeicher (Flash) als auch im Arbeitsspeicher (RAM) gespeichert. Die Startwerte (Anfangswerte) werden immer im Ladespeicher gespeichert und die aktuellen Werte werden immer im Arbeitsspeicher gespeichert. Mit READ\_DBL können Sie einen Satz Startwerte aus dem Ladespeicher in die aktuellen Werte eines DBs im Arbeitsspeicher kopieren, der von Ihrem Programm referenziert wird. Mit WRIT\_DBL können Sie die im internen Ladespeicher oder auf einer Memory Card gespeicherten aktuellen Werte im Arbeitsspeicher aktualisieren.

---

#### Hinweis

##### **Vermeiden Sie übermäßige Schreibvorgänge mit WRIT\_DBL in den Flash-Speicher**

Die Anweisung WRIT\_DBL führt Schreibvorgänge im Flash-Speicher durch (im internen Ladespeicher oder auf einer Memory Card). WRIT\_DBL sollte lediglich für selten vorkommende Aktualisierungen wie Änderungen im Produktionsprozess genutzt werden.

---

Die von READ\_DBL und WRIT\_DBL verwendeten Datenbausteine müssen zuvor von STEP 7 erstellt worden sein, damit Sie diese Anweisungen nutzen können. Wenn der Quell-DB als Baustein vom Typ "Standard" erstellt wurde, muss der Ziel-DB ebenfalls vom Typ "Standard" sein. Wenn der Quelldatenbaustein als Baustein vom Typ "Optimiert" erstellt wurde, muss der Zieldatenbaustein ebenfalls vom Typ "Optimiert" sein.

Wenn es sich um Standard-DBs handelt, können Sie entweder einen Variablennamen oder einen P#-Wert angeben. Mit dem P#-Wert können Sie eine beliebige Anzahl von Elementen der angegebenen Größe (Byte, Word oder DWord) angeben und kopieren. So können Sie einen Teil eines DBs oder den gesamten DB kopieren. Handelt es sich um optimierte DBs, können Sie lediglich einen Variablennamen angeben. Den Operator P# können Sie nicht verwenden. Wenn Sie für einen Standard-DB oder einen optimierten DB (oder für andere Arten von Arbeitsspeichern) einen Variablennamen angeben, wird das, was von diesem Variablennamen referenziert wird, kopiert. Hierbei kann es sich um einen benutzerdefinierten Typ, ein Array oder ein grundlegendes Element handeln. Der Datentyp Struct kann von diesen Anweisungen nur verwendet werden, wenn es sich um einen Standard-DB und nicht um einen optimierten DB handelt. Sie müssen einen benutzerdefinierten Datentyp (UDT) verwenden, wenn es sich um eine Struktur im optimierten Speicher handelt. Nur mit einem benutzerdefinierten Typ wird sichergestellt, dass die "Datentypen" sowohl für die Quellstruktur als auch für die Zielstruktur identisch sind.

---

#### Hinweis

##### **Struktur (Datentyp Struct) in einem "optimierten" DB verwenden**

Wenn Sie einen Datentyp Struct mit "optimierten" DBs verwenden, müssen Sie zunächst einen benutzerdefinierten Datentyp (UDT) für den Datentyp Struct erstellen. Dann konfigurieren Sie den Quell- und Ziel-DB mit dem Datentyp UDT. Der Datentyp UDT gewährleistet, dass die Datentypen im Datentyp Struct für beide DBs konsistent bleiben.

Bei "Standard"-DBs verwenden Sie den Datentyp Struct, ohne einen Datentyp UDT zu erstellen.

---

READ\_DBL und WRIT\_DBL werden asynchron zum Programmzyklus ausgeführt. Die Verarbeitung erstreckt sich über mehrere Aufrufe von READ\_DBL und WRIT\_DBL. Sie starten den DB-Übertragungsauftrag durch Aufruf von REQ = 1 und überwachen anschließend die Ausgänge BUSY und RET\_VAL, um zu ermitteln, wann die Datenübertragung beendet und korrekt ist.

Um Datenkonsistenz sicherzustellen, ändern Sie den Zielbereich während der Bearbeitung von READ\_DBL bzw. den Quellbereich während der Bearbeitung von WRIT\_DBL nicht (d.h. solange der Parameter BUSY WAHR ist).

Einschränkungen bei den Parametern SRCBLK und DSTBLK:

- Ein Datenbaustein muss, damit er referenziert werden kann, zuvor erstellt worden sein.
- Die Länge eines VARIANT-Pointers vom Typ BOOL muss durch 8 teilbar sein.
- Die Länge eines VARIANT-Pointers vom Typ STRING muss in den Quell- und Ziel-Pointern identisch sein.

### Informationen zu Rezepten und zur Maschineneinrichtung

Mit den Anweisungen READ\_DBL und WRIT\_DBL können Sie Rezepte oder Informationen für die Maschineneinrichtung verwalten. Dies ist prinzipiell eine weitere Vorgehensweise, um für Werte, die sich nicht sehr häufig ändern, remanente Daten zu speichern. Sie sollten jedoch die Anzahl der Schreibvorgänge begrenzen, um den Flash-Speicher nicht übermäßig zu beanspruchen. Sie können auf diese Weise den Umfang des remanenten Speichers über den normalen Umfang der remanenten Daten hinaus erweitern, zumindest für Werte, die sich nicht häufig ändern. Sie können mit der Anweisung WRIT\_DBL Rezeptinformationen oder Informationen für die Maschineneinrichtung aus dem Arbeitsspeicher im Ladespeicher speichern, und Sie können diese Informationen mit der Anweisung READ\_DBL wieder aus dem Ladespeicher in den Arbeitsspeicher abrufen.

Tabelle 8- 143    Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
0081	Warnung: Der Quellbereich ist kleiner als der Zielbereich. Die Quelldaten werden vollständig kopiert, wobei die zusätzlichen Bytes im Zielbereich nicht verändert werden.
7000	Aufruf mit REQ = 0: BUSY = 0
7001	Erster Aufruf mit REQ = 1 (in Bearbeitung): BUSY = 1
7002	N <sup>ter</sup> Aufruf (in Bearbeitung): BUSY = 1
8051	Typfehler Datenbaustein
8081	Der Quellbereich ist größer als der Zielbereich. Der Zielbereich wird vollständig belegt und die verbleibenden Bytes der Quelle werden ignoriert.
8251	Typfehler Quelldatenbaustein
82B1	Fehlender Quelldatenbaustein
82C0	Der Quell-DB wird von einer anderen Anweisung oder Kommunikationsfunktion bearbeitet.
8551	Typfehler Zieldatenbaustein
85B1	Fehlender Zieldatenbaustein
85C0	Der Ziel-DB wird von einer anderen Anweisung oder Kommunikationsfunktion bearbeitet.
80C3	Mehr als 50 READ_DBL- oder 50 WRIT_DBL-Anweisungen befinden sich derzeit in der Warteschlange für die Ausführung.

## 8.9 Gemeinsame Fehlercodes für die erweiterten Anweisungen

Tabelle 8- 144 Gemeinsame Bedingungscode für die erweiterten Anweisungen

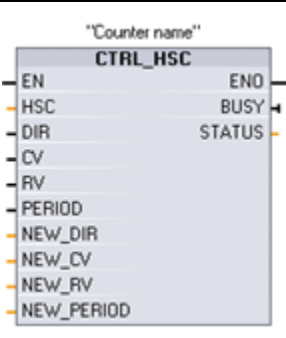
Bedingungscode (W#16#...) <sup>1</sup>	Beschreibung
8022	Bereich für Eingang zu klein
8023	Bereich für Ausgang zu klein
8024	Unzulässiger Eingangsbereich
8025	Unzulässiger Ausgangsbereich
8028	Unzulässige Eingangsbitzuweisung
8029	Unzulässige Ausgangsbitzuweisung
8030	Ausgangsbereich ist ein schreibgeschützter DB.
803A	DB ist nicht vorhanden.

<sup>1</sup> Wenn einer dieser Fehler während der Ausführung eines Codebausteins auftritt, geht die CPU in den Betriebszustand STOP, es sei denn, Sie nutzen eine der Anweisungen GetError oder GetErrorID in dem Codebaustein und erreichen eine programmierte Reaktion auf den Fehler.

# Technologieanweisungen

## 9.1 Schneller Zähler

Tabelle 9- 1 Anweisung CTRL\_HSC

KOP/FUP	SCL	Beschreibung
	<pre>"CTRL_HSC_0_DB" (   hsc:=_hw_hsc_in_,   dir:=_bool_in_,   cv:=_bool_in_,   rv:=_bool_in_,   period:=_bool_in_,   new_dir:=_int_in_,   new_cv:=_int_in_,   new_rv:=_dint_in_,   new_period:=_int_in_,   busy:=_bool_out_,   status:=_word_out_);</pre>	<p>Jede Anweisung CTRL_HSC nutzt eine in einem DB abgelegte Struktur, um Daten zu speichern. Sie weisen den DB zu, wenn Sie die Anweisung CTRL_HSC im Editor einfügen.</p>

- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "CTRL\_HSC\_0\_DB" der Name des Instanz-DBs.

Tabelle 9- 2 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
HSC	IN	HW_HSC	HSC-Kennung
DIR <sup>1, 2</sup>	IN	Bool	1 = Neue Richtung anfordern
CV <sup>1</sup>	IN	Bool	1 = Anforderung zum Setzen eines neuen Zählwerts
RV <sup>1</sup>	IN	Bool	1 = Anforderung zum Setzen eines neuen Referenzwerts
PERIOD <sup>1</sup>	IN	Bool	1 = Anforderung zum Setzen eines neuen Zeitintervalls (nur bei Frequenzmessung)
NEW_DIR	IN	Int	Neue Richtung: 1= vorwärts, -1= rückwärts
NEW_CV	IN	DInt	Neuer Zählwert
NEW_RV	IN	DInt	Neuer Referenzwert
NEW_PERIOD	IN	Int	Neuer Zeitintervallwert in Sekunden: 0,01, 0,1 oder 1 (nur bei Frequenzmessung)
BUSY <sup>3</sup>	OUT	Bool	Funktion besetzt
STATUS	OUT	Word	Ausführungsbedingung

- Wird keine Aktualisierung eines Parameterwerts angefordert, so werden die entsprechenden Eingangswerte ignoriert.
- Der Parameter DIR ist nur gültig, wenn für die konfigurierte Zählrichtung die Programmsteuerung (interne Richtungssteuerung) vorgegeben ist. Sie legen fest, wie dieser Parameter in der HSC-Gerätekonfiguration genutzt wird.
- Bei einem HSC in der CPU oder im SB hat der Parameter BUSY immer den Wert 0.

Sie konfigurieren die Parameter für jeden HSC in der Gerätekonfiguration für die CPU: Zählermodus, E/A-Anschaltungen, Alarmzuweisung und Betrieb als schneller Zähler oder als Gerät für die Impulsfrequenzmessung.

Einige der Parameter für den HSC können von Ihrem Anwenderprogramm geändert werden, um die Programmsteuerung für den Zählvorgang vorzugeben:

- Setzen der Zählrichtung auf einen Wert NEW\_DIR
- Setzen des aktuellen Zählwerts auf einen neuen Wert NEW\_CV
- Setzen des Referenzwerts auf einen neuen Wert NEW\_RV
- Setzen des Zeitintervallwerts (nur bei Frequenzmessung) auf einen neuen Wert NEW\_PERIOD

Sind die folgenden Booleschen Merker auf 1 gesetzt, wenn die Anweisung CTRL\_HSC ausgeführt wird, so wird der entsprechende Wert NEW\_xxx in den Zähler geladen. Mehrere Anforderungen (mehrere Merker sind gleichzeitig gesetzt) werden in einer Ausführung der Anweisung CTRL\_HSC verarbeitet.

- DIR = 1 ist eine Anforderung zum Laden eines Werts NEW\_DIR, 0 = keine Änderung
- CV = 1 ist eine Anforderung zum Laden eines Werts NEW\_CV, 0 = keine Änderung
- RV = 1 ist eine Anforderung zum Laden eines Werts NEW\_RV, 0 = keine Änderung
- PERIOD = 1 ist eine Anforderung zum Laden eines Werts NEW\_PERIOD, 0 = keine Änderung

Die Anweisung CTRL\_HSC wird typischerweise in einem Prozessalarm-OB platziert, der ausgeführt wird, wenn das Prozessalarmereignis des Zählers ausgelöst wird. Wird zum Beispiel der Zählerinterrupt durch ein Ereignis CV=RV ausgelöst, so führt ein Prozessalarm-OB die Anweisung CTRL\_HSC aus und kann den Referenzwert durch Laden eines Werts NEW\_RV ändern.

Der aktuelle Zählwert ist in den Parametern für CTRL\_HSC nicht vorgesehen. Die Adresse des Prozessabbilds, in der der aktuelle Zählwert gespeichert wird, wird bei der Hardwarekonfiguration dem schnellen Zähler zugewiesen. Sie können den Zählwert über die Programmlogik direkt auslesen. Der an Ihr Programm ausgegebene Wert ist ein korrekter Zählwert für den Moment, in dem der Zähler gelesen wurde. Der Zähler setzt die Zählung schneller Ereignisse fort. Der tatsächliche Zählwert kann sich deshalb ändern, bevor Ihr Programm einen Prozess mit einem alten Zählwert beendet.

**Bedingungscode:** Bei einem Fehler wird ENO auf 0 gesetzt und der Ausgang STATUS enthält einen Bedingungscode.

Tabelle 9- 3 STATUS-Werte (W#16#)

STATUS	Beschreibung
0	Kein Fehler
80A1	HSC-Kennung adressiert keinen HSC
80B1	Ungültiger Wert in NEW_DIR
80B2	Ungültiger Wert in NEW_CV
80B3	Ungültiger Wert in NEW_RV
80B4	Ungültiger Wert in NEW_PERIOD

STATUS	Beschreibung
80C0	Mehrmaliger Zugriff auf den schnellen Zähler
80D0	Schneller Zähler (HSC) nicht in der CPU-Hardwarekonfiguration aktiviert

### 9.1.1 Funktionsweise von schnellen Zählern

Der schnelle Zähler (HSC) zählt Ereignisse, die schneller als die Ausführungsrate des OBs auftreten. Wenn die zu zählenden Ereignisse innerhalb der Ausführungsrate des OBs auftreten, können Sie die Zähleranweisungen CTU, CTD oder CTUD verwenden. Wenn die Ereignisse schneller als die Ausführungsrate des OBs auftreten, verwenden Sie den HSC. Mit der Anweisung CTRL\_HSC kann Ihr Anwenderprogramm programmatisch einige der HSC-Parameter ändern.

Beispiel: Sie können den HSC als Eingang für einen Winkelschrittgeber nutzen. Der Winkelschrittgeber sorgt für eine bestimmte Anzahl von Zählwerten pro Umdrehung sowie für einen Rücksetzimpuls einmal pro Umdrehung. Der bzw. die Taktgeber und der Rücksetzimpuls des Winkelschrittgebers liefern die Eingänge für den schnellen Zähler.

Der erste von mehreren voreingestellten Werten wird in den schnellen Zähler geladen. Die Ausgänge werden für die Zeitspanne aktiviert, während der der aktuelle Wert des Zählers kleiner als der voreingestellte Wert ist. Der HSC löst einen Alarm aus, wenn der aktuelle Wert des Zählers gleich dem voreingestellten Wert ist, wenn der Zähler zurückgesetzt wird oder wenn ein Richtungswechsel auftritt.

Wenn der aktuelle Wert gleich dem voreingestellten Wert ist und es zu einem Alarmereignis kommt, dann wird ein neuer voreingestellter Wert geladen und der nächste Signalzustand für die Ausgänge gesetzt. Tritt ein Alarmereignis auf, weil der Zähler zurückgesetzt wird, dann werden der erste voreingestellte Wert und die ersten Signalzustände der Ausgänge gesetzt und der Zyklus wiederholt.

Da die Alarme in einer sehr viel geringeren Geschwindigkeit auftreten als der schnelle Zähler zählt, kann eine präzise Steuerung der schnellen Anweisungen mit relativ geringem Einfluss auf den Zyklus der CPU implementiert werden. Da Sie Alarme bestimmten Interruptprogrammen zuordnen können, kann jede neue Voreinstellung in einem getrennten Interruptprogramm geladen werden, damit so der Zustand einfach gesteuert werden kann. (Sie können alternativ auch alle Alarmereignisse in einem einzigen Interruptprogramm bearbeiten.)

Tabelle 9-4 Maximale Frequenz (kHz)

HSC		Einphasenzähler	Zweiphasenzähler und A/B-Zähler
HSC1	CPU	100 kHz	80 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC2	CPU	100 kHz	80 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC3	CPU	100 kHz	80 kHz
HSC4	CPU	30 kHz	20 kHz

HSC		Einphasenzähler	Zweiphasenzähler und A/B-Zähler
HSC5	CPU	30 kHz	20 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC6	CPU	30 kHz	20 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz

### Funktionalität für den HSC auswählen

Alle HSCs arbeiten in der gleichen Zählerart auf die gleiche Weise. Es gibt vier grundlegende Arten von schnellen Zählern:

- Einphasenzähler mit interner Richtungssteuerung
- Einphasenzähler mit externer Richtungssteuerung
- Zweiphasenzähler mit 2 Takteingängen
- A/B-Zähler

Sie können jeden HSC-Typ mit oder ohne Rücksetzeingang verwenden. Wenn Sie den Rücksetzeingang aktivieren (mit einigen Einschränkungen, siehe folgende Tabelle), wird der aktuelle Wert zurückgesetzt. Er bleibt so lange zurückgesetzt, bis Sie den Rücksetzeingang deaktivieren.

- Frequenzfunktion: Der HSC kann in bestimmten Betriebsarten so konfiguriert werden (Zählart), dass er eine Frequenz statt der aktuellen Impulszahl ausgibt. Es gibt drei verschiedene Frequenzmesszeiten: 0,01, 0,1 oder 1,0 Sekunden.

Die Frequenzmesszeit legt fest, wie oft der HSC einen neuen Frequenzwert berechnet und ausgibt. Die ausgegebene Frequenz ist ein Mittelwert, der anhand der Gesamtzahl der Zählwerte im letzten Messzeitraum berechnet wird. Ändert sich die Frequenz schnell, ist der ausgegebene Wert ein Zwischenergebnis zwischen der höchsten und der niedrigsten während des Messzeitraums aufgetretenen Frequenz. Die Frequenz wird immer in Hertz (Impulse pro Sekunde) angegeben, unabhängig von der eingestellten Frequenzmesszeit.

- Zählerarten und Zählereingänge: Die folgende Tabelle zeigt die Eingänge, die für Funktionen wie Taktgeber, Richtungssteuerung und Zurücksetzen des HSC verwendet werden.

Ein Eingang kann nicht für zwei verschiedene Funktionen verwendet werden. Wird ein Eingang jedoch nicht von der aktuellen Zählerart des definierten schnellen Zählers benötigt, kann er für andere Zwecke genutzt werden. Ist HSC1 in einer Betriebsart, in der die integrierten Eingänge, aber nicht der externe Rücksetzeingang (E0.3) verwendet wird, so kann E0.3 beispielsweise für Flankeninterrupts oder für HSC2 belegt werden.

Tabelle 9- 5 Zählarten für den HSC

Typ	Eingang 1	Eingang 2	Eingang 3	Funktion
Einphasenzähler mit interner Richtungssteuerung	Takt	(Optional: Richtung)	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert



Typ	Eingang 1	Eingang 2	Eingang 3	Funktion
Einphasenzähler mit externer Richtungssteuerung	Takt	Richtung	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert
Zweiphasenzähler mit 2 Takteingängen	Takt vorwärts	Takt rückwärts	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert
A/B-Zähler	Phase A	Phase B	-	Zählwert oder Frequenz
			Zurücksetzen <sup>1</sup>	Zählwert

<sup>1</sup> Bei einem Winkelschrittgeber: Phase Z, Referenzpunkt

## Eingangsadressen für den HSC

### Hinweis

Die von schnellen Zählern verwendeten digitalen E/A werden während der Gerätekonfiguration zugewiesen. Wenn diesen Funktionen digitale E/A zugewiesen werden, können die Werte der Adressen der zugewiesenen E/A nicht durch die Forcefunktion einer Beobachtungstabelle geändert werden.

Wenn Sie die CPU konfigurieren, können Sie jeden HSC aktivieren und konfigurieren. Die CPU weist die Eingangsadressen für jeden HSC automatisch entsprechend der Konfiguration zu. (Bei einigen der HSCs können Sie auswählen, ob Sie die integrierten Eingänge der CPU oder die Eingänge eines SB nutzen möchten.)

### ACHTUNG

Wie Sie in den folgenden Tabellen sehen, überschneiden sich die Standardzuweisungen der optionalen Signale der verschiedenen HSCs. Zum Beispiel wird für das optionale externe Rücksetzen von HSC 1 der gleiche Eingang verwendet wie für einen der Eingänge von HSC 2.

Stellen Sie stets sicher, dass Sie Ihre HSCs so konfigurieren, dass **kein** Eingang von zwei HSCs verwendet wird.

Die folgende Tabelle zeigt die HSC-Eingangszuweisungen für die integrierten E/A der CPU 1211C und eines SB. (Wenn das SB nur zwei Eingänge hat, sind nur die Eingänge 4.0 und 4.1 verfügbar.)

- Bei Einphasenzählern: C ist der Zähl Eingang, [d] ist der optionale Richtungseingang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei Zweiphasenzählern: CU ist der Vorwärtzähl Eingang, CD ist der Rückwärtzähl Eingang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei A/B-Zählern: A ist der Zähl Eingang A, B ist der Zähl Eingang B und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)

Tabelle 9- 6 HSC-Eingangszuweisungen bei der CPU 1211C

HSC		Integrierter CPU-Eingang (0.x)						SB-Eingang (Standard 4.x) <sup>3</sup>			
		0	1	2	3	4	5	0	1	2	3
HSC 1 <sup>1</sup>	Einphasig	C	[d]		[R]			C	[d]		[R]
	Zweiphasig	CU	CD		[R]			CU	CD		[R]
	A/B-Zähler	A	B		[R]			A	B		[R]
HSC 2 <sup>1</sup>	Einphasig		[R]	C	[d]				[R]	C	[d]
	Zweiphasig		[R]	CU	CD				[R]	CU	CD
	A/B-Zähler		[R]	A	B				[R]	A	B
HSC 3	Einphasig					C	[d]				
	Zweiphasig					CU	CD				
	A/B-Zähler					A	B				
HSC 5 <sup>2</sup>	Einphasig							C	[d]		[R]
	Zweiphasig							CU	CD		[R]
	A/B-Zähler							A	B		[R]
HSC 6 <sup>2</sup>	Einphasig								[R]	C	[d]
	Zweiphasig								[R]	CU	CD
	A/B-Zähler								[R]	A	B

<sup>1</sup> HSC 1 und HSC 2 können für integrierte Eingänge oder für ein SB konfiguriert werden.

<sup>2</sup> HSC 5 und HSC 6 sind nur bei einem SB verfügbar. HSC 6 ist nur bei einem SB mit 4 Eingängen verfügbar.

<sup>3</sup> Ein SB mit nur zwei Digitaleingängen bietet nur die Eingänge 4.0 und 4.1.

Die folgende Tabelle zeigt die HSC-Eingangszuweisungen für die integrierten E/A der CPU 1212C und eines SB. (Wenn das SB nur zwei Eingänge hat, sind nur die Eingänge 4.0 und 4.1 verfügbar.)

- Bei Einphasenzählern: C ist der Zählengang, [d] ist der optionale Richtungseingang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei Zweiphasenzählern: CU ist der Vorwärtzählengang, CD ist der Rückwärtzählengang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei A/B-Zählern: A ist der Zählengang A, B ist der Zählengang B und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)

Tabelle 9- 7 HSC-Eingangszuweisungen bei der CPU 1212C

HSC		Integrierter CPU-Eingang (0.x)								SB-Eingang (4.x) <sup>3</sup>			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1 <sup>1</sup>	Einphasig	C	[d]		[R]					C	[d]		[R]
	Zweiphasig	CU	CD		[R]					CU	CD		[R]
	A/B-Zähler	A	B		[R]					A	B		[R]
HSC 2 <sup>1</sup>	Einphasig		[R]	C	[d]						[R]	C	[d]
	Zweiphasig		[R]	CU	CD						[R]	CU	CD

HSC		Integrierter CPU-Eingang (0.x)							SB-Eingang (4.x) <sup>3</sup>				
		0	1	2	3	4	5	6	7	0	1	2	3
	A/B-Zähler		[R]	A	B						[R]	A	B
HSC 3	Einphasig					C	[d]		[R]				
	Zweiphasig					CU	CD		[R]				
	A/B-Zähler					A	B		[R]				
HSC 4	Einphasig						[R]	C	[d]				
	Zweiphasig						[R]	CU	CD				
	A/B-Zähler						[R]	A	B				
HSC 5 <sup>2</sup>	Einphasig									C	[d]		[R]
	Zweiphasig									CU	CD		[R]
	A/B-Zähler									A	B		[R]
HSC 6 <sup>2</sup>	Einphasig										[R]	C	[d]
	Zweiphasig										[R]	CU	CD
	A/B-Zähler										[R]	A	B

- 1 HSC 1 und HSC 2 können für integrierte Eingänge oder für ein SB konfiguriert werden.
- 2 HSC 5 und HSC 6 sind nur bei einem SB verfügbar. HSC 6 ist nur bei einem SB mit 4 Eingängen verfügbar.
- 3 Ein SB mit nur zwei Digitaleingängen bietet nur die Eingänge 4.0 und 4.1.

Die folgenden zwei Tabellen zeigen die HSC-Eingangszuweisungen für die integrierten E/A der CPU 1214C und für ein ggf. installiertes optionales SB.

- Bei Einphasenzählern: C ist der Zähleingang, [d] ist der optionale Richtungseingang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei Zweiphasenzählern: CU ist der Vorwärtzähleingang, CD ist der Rückwärtzähleingang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei A/B-Zählern: A ist der Zähleingang A, B ist der Zähleingang B und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)

Tabelle 9- 8 HSC-Eingangszuweisungen für die CPU 1214C und CPU 1215C (nur integrierte Eingänge)

HSC		Digitaleingang 0 (Standard: 0.x)							Digitaleingang 1 (Standard: 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1 <sup>1</sup>	Einphasig	C	[d]		[R]										
	Zweiphasig	CU	CD		[R]										
	A/B-Zähler	A	B		[R]										
HSC 2 <sup>1</sup>	Einphasig		[R]	C	[d]										
	Zweiphasig		[R]	CU	CD										
	A/B-Zähler		[R]	A	B										
HSC 3	Einphasig					C	[d]		[R]						
	Zweiphasig					CU	CD		[R]						
	A/B-Zähler					A	B		[R]						

HSC		Digitaleingang 0 (Standard: 0.x)							Digitaleingang 1 (Standard: 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 4	Einphasig						[R]	C	[d]						
	Zweiphasig						[R]	CU	CD						
	A/B-Zähler						[R]	A	B						
HSC 5 <sup>1</sup>	Einphasig									C	[d]	[R]			
	Zweiphasig									CU	CD	[R]			
	A/B-Zähler									A	B	[R]			
HSC 6 <sup>1</sup>	Einphasig												C	[d]	[R]
	Zweiphasig												CU	CD	[R]
	A/B-Zähler												A	B	[R]

<sup>1</sup> HSC 1, HSC 2, HSC 5 und HSC 6 können für integrierte Eingänge oder für ein SB konfiguriert werden.

Tabelle 9-9 HSC-Eingangszuweisungen bei SBs

HSC <sup>1</sup>		SB-Eingänge (Standard: 4.x) <sup>2</sup>			
		0	1	2	3
HSC 1	Einphasig	C	[d]		[R]
	Zweiphasig	CU	CD		[R]
	A/B-Zähler	A	B		[R]
HSC 2	Einphasig		[R]	C	[d]
	Zweiphasig		[R]	CU	CD
	A/B-Zähler		[R]	A	B
HSC 5	Einphasig	C	[d]		[R]
	Zweiphasig	CU	CD		[R]
	A/B-Zähler	A	B		[R]
HSC 6	Einphasig		[R]	C	[d]
	Zweiphasig		[R]	CU	CD
	A/B-Zähler		[R]	A	B

<sup>1</sup> Bei der CPU 1214C: HSC 1, HSC 2, HSC 5 und HSC 6 können für integrierte Eingänge oder für ein SB konfiguriert werden.

<sup>2</sup> Ein SB mit nur zwei Digitaleingängen bietet nur die Eingänge 4.0 und 4.1.

## Zugreifen auf den aktuellen Wert des HSC

### Hinweis

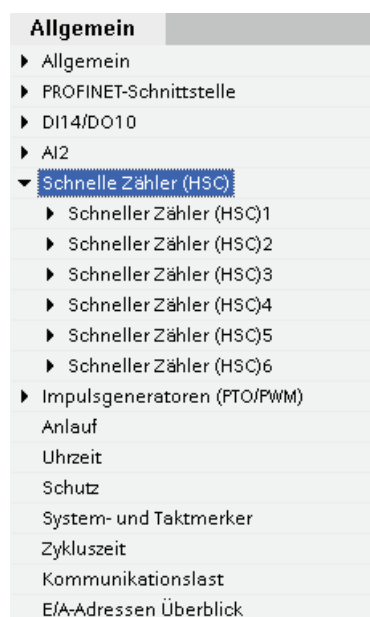
Wenn Sie einen Impulsgenerator als PTO aktivieren, wird diesem PTO ein entsprechender HSC zugewiesen. HSC1 wird PTO1 zugewiesen, und HSC2 wird PTO2 zugewiesen. Der zugewiesene HSC gehört vollständig dem PTO-Kanal, und der gewöhnliche Ausgang des HSC ist deaktiviert. Der HSC-Wert wird nur für die interne Funktionalität verwendet. Sie können den aktuellen Wert (z. B. in ID1000) nicht überwachen, wenn Impulse auftreten.

Die CPU speichert den aktuellen Wert jedes HSC in der Adresse eines Eingangs (E). Die folgende Tabelle zeigt die Standardadressen für den aktuellen Wert jedes HSC. Sie können die E-Adresse für den aktuellen Wert ändern, indem Sie die Eigenschaften der CPU in der Gerätekonfiguration ändern.

Tabelle 9- 10 Aktueller Wert des HSC

HSC	Datentyp	Standardadresse
HSC1	DInt	ED1000
HSC2	DInt	ED1004
HSC3	DInt	ED1008
HSC4	DInt	ED1012
HSC5	DInt	ED1016
HSC6	DInt	ED1020

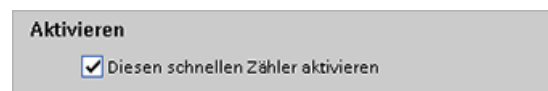
## 9.1.2 Konfiguration eines schnellen Zählers



Die CPU ermöglicht das Konfigurieren von maximal 6 schnellen Zählern. Unter den "Eigenschaften" der CPU können Sie die Parameter für jeden schnellen Zähler einrichten.

Mit der Anweisung CTRL\_HSC in Ihrem Anwenderprogramm können Sie die Funktionsweise des HSC steuern.

Sie aktivieren den spezifischen HSC durch Auswahl der Option "Aktivieren" für den HSC.



### Hinweis

Wenn Sie den schnellen Zähler aktivieren und Eingänge für ihn auswählen, werden für die Einstellungen des Eingangsfilters für diese Eingänge 800 ns festgelegt. Jeder Eingang hat eine Filterkonfiguration, die für alle Zwecke gilt: Prozesseingänge, Alarme, Impulsabgriff und HSC-Eingänge.

**! WARNUNG**

Wenn die bisherige Einstellung der Filterzeit eines Digitaleingangskanals verändert wird, muss möglicherweise bis zu 20,0 ms lang ein neuer Eingangswert mit Pegel "0" vorhanden sein, damit der Filter vollständig auf neue Eingänge reagiert. Während dieses Zeitraums werden kurze Impulse mit Pegel "0", die kürzer als 20,0 ms sind, möglicherweise nicht erkannt oder gezählt.

Das Verändern der Filterzeiten kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.

Um sicherzustellen, dass eine neue Filterzeit sofort wirksam wird, müssen Sie die CPU aus- und wieder einschalten.

Nach der Aktivierung des HSC konfigurieren Sie die anderen Parameter wie Zählerfunktion, Anfangswerte, Rücksetzoptionen und Interruptereignisse.



Weitere Informationen zum Konfigurieren des HSC finden Sie im Abschnitt zum Konfigurieren der CPU (Seite 133).

## 9.2 PID-Regelung

STEP 7 bietet die folgenden PID-Anweisungen für die S7-1200 CPU:

- Die Anweisung PID\_Compact dient zum Regeln technischer Prozesse mit kontinuierlichen Eingangs- und Ausgangsvariablen.
- Die Anweisung PID\_3Step dient zum Regeln von motorbetätigten Geräten wie Ventilen, die digitale Signale zum Öffnen und Schließen benötigen.

---

### Hinweis

Änderungen, die Sie an der PID-Konfiguration vornehmen und im Betriebszustand RUN ins Zielsystem laden, werden erst dann wirksam, wenn Sie die CPU von STOP in RUN versetzen.

---

Beide PID-Anweisungen (PID\_3Step und PID\_Compact) können den P-, I- und D-Anteil während des Anlaufs berechnen (sofern die "Erstoptimierung" konfiguriert ist). Sie können die Anweisung auch für die "Feineinstellung" konfigurieren, um die Parameter zu optimieren. Sie brauchen die Parameter nicht manuell festzulegen.

---

**Hinweis**

**Führen Sie die PID-Anweisung in einem wiederholten konstanten Zeitintervall der Abtastzeit aus (nach Möglichkeit in einem Zyklus-OB).**

Weil der PID-Regler eine bestimmte Zeit benötigt, um auf Änderungen des Stellwerts zu reagieren, berechnen Sie den Ausgangswert nicht in jedem Zyklus. Führen Sie die PID-Anweisung nicht im Zyklus-OB des Hauptprogramms aus (z. B. OB 1).

---

Die Abtastzeit des PID-Algorithmus stellt die Zeit zwischen zwei Berechnungen des Ausgangswerts (Stellwerts) dar. Der Ausgangswert wird während der Selbsteinstellung berechnet und auf ein Vielfaches der Zykluszeit gerundet. Alle anderen Funktionen der PID-Anweisung werden bei jedem Aufruf ausgeführt.

## PID-Algorithmus

Der PID-Regler (Proportional/Integral/Differential) misst das Zeitintervall zwischen zwei Aufrufen und wertet dann die Ergebnisse aus, um die Abtastzeit zu überwachen. Bei jedem Wechsel des Betriebszustands sowie beim ersten Anlauf wird ein Mittelwert der Abtastzeit errechnet. Dieser Wert dient als Referenzwert für die Überwachungsfunktion und zur Berechnung. Bei der Überwachung wird die aktuelle Messzeit zwischen zwei Aufrufen und der Mittelwert der für den Regler festgelegten Abtastzeit überwacht.

Der Ausgangswert des PID-Reglers besteht aus drei Anteilen:

- P (Proportional): Bei Berechnung mit P-Anteil ist der Ausgangswert proportional zur Differenz zwischen dem Sollwert und dem Prozesswert (Eingangswert).
- I (Integral): Bei Berechnung mit I-Anteil steigt der Ausgangswert proportional zur Dauer der Differenz zwischen dem Sollwert und dem Prozesswert (Eingangswert), um letztlich die Differenz zu korrigieren.
- D (Differential): Bei Berechnung mit D-Anteil steigt der Ausgangswert als Funktion der zunehmenden Änderungsgeschwindigkeit der Differenz zwischen dem Sollwert und dem Prozesswert (Eingangswert). Der Ausgangswert wird so schnell wie möglich an den Sollwert angeglichen.

Der PID-Regler berechnet den Ausgangswert für die Anweisung PID\_Compact anhand folgender Formel.

$$y = K_p \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Ausgangswert	x	Prozesswert
w	Sollwert	s	Laplace-Operator
K <sub>p</sub>	Proportionale Verstärkung (P-Anteil)	a	Koeffizient für den Differenzierverszug (D-Anteil)
T <sub>i</sub>	Integralzeit (I-Anteil)	b	Gewichtung des Proportionalanteils (P-Anteil)
T <sub>D</sub>	Differentialzeit (D-Anteil)	c	Gewichtung des Differentialanteils (D-Anteil)

Der PID-Regler berechnet den Ausgangswert für die Anweisung PID\_3Step anhand folgender Formel.

$$\Delta y = K_p \cdot s \cdot \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Ausgangswert	x	Prozesswert
w	Sollwert	s	Laplace-Operator
K <sub>p</sub>	Proportionale Verstärkung (P-Anteil)	a	Koeffizient für den Differenzierverszug (D-Anteil)
T <sub>i</sub>	Integralzeit (I-Anteil)	b	Gewichtung des Proportionalanteils (P-Anteil)
T <sub>D</sub>	Differentialzeit (D-Anteil)	c	Gewichtung des Differentialanteils (D-Anteil)

### 9.2.1 Anweisung PID und Technologieobjekt einfügen

STEP 7 bietet zwei Anweisungen für den PID-Regler:

- Die Anweisung PID\_Compact und das zugehörige Technologieobjekt bieten einen universalen PID-Regler mit Einstellung. Das Technologieobjekt enthält alle Einstellungen für den Regelkreis.
- Die Anweisung PID\_3Step und das zugehörige Technologieobjekt bieten einen PID-Regler mit bestimmten Einstellungen für motorbetätigte Ventile. Das Technologieobjekt enthält alle Einstellungen für den Regelkreis. Der Regler PID\_3Step bietet zwei zusätzliche Boolesche Ausgänge.

Nach dem Anlegen des Technologieobjekts müssen Sie die Parameter konfigurieren (Seite 384). Sie passen auch die Parameter für die Selbsteinstellung an ("Erstoptimierung" während des Anlaufs oder manuelle "Feineinstellung"), um den PID-Regler in Betrieb zu nehmen (Seite 386).



Tabelle 9- 11 Anweisung PID und Technologieobjekt einfügen

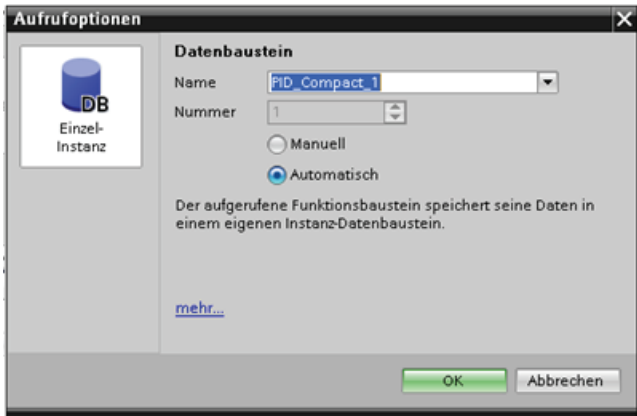
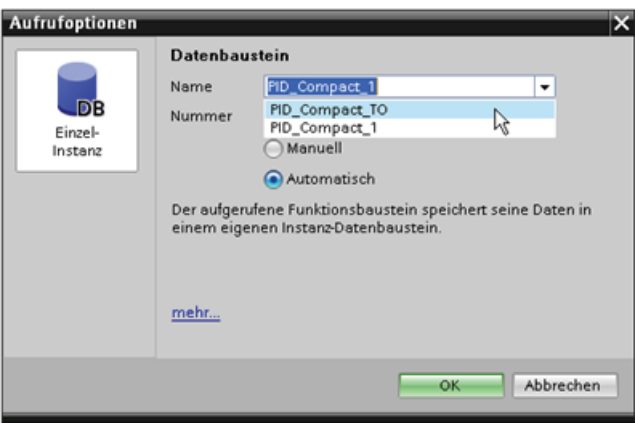
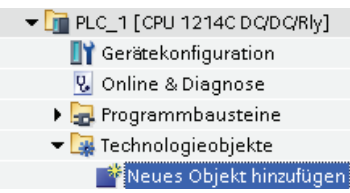
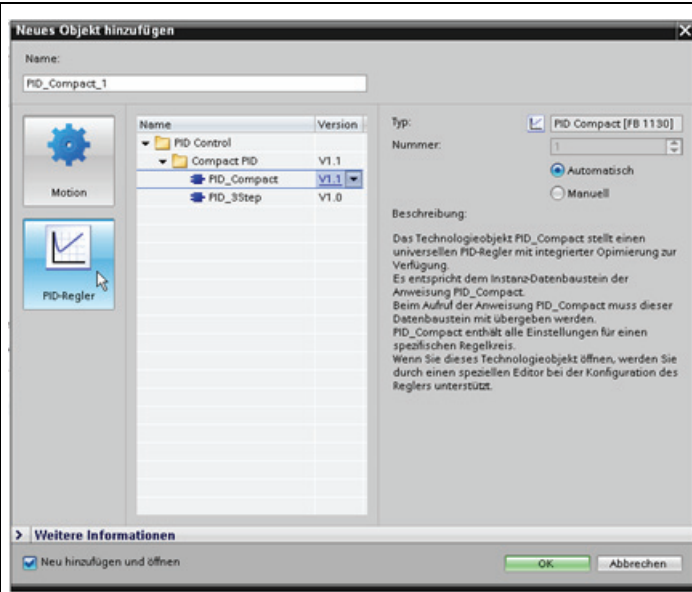
<p>Wenn Sie eine PID-Anweisung in Ihr Anwenderprogramm einfügen, legt STEP 7 automatisch ein Technologieobjekt und einen Instanz-DB für die Anweisung an. Der Instanz-DB enthält alle Parameter für die PID-Anweisung. Jede PID-Anweisung benötigt einen eigenen, eindeutigen Instanz-DB, um ordnungsgemäß zu funktionieren.</p> <p>Nach dem Einfügen der PID-Anweisung und dem Erstellen des Technologieobjekts sowie des Instanz-DBs konfigurieren Sie die Parameter für das Technologieobjekt (Seite 384).</p>	
---	--

Tabelle 9- 12 (Optional) Technologieobjekt in der Projektnavigation anlegen

<p>Sie können Technologieobjekte für Ihr Projekt auch vor dem Einfügen der PID-Anweisung anlegen. Wenn Sie das Technologieobjekt vor dem Einfügen der PID-Anweisung in Ihr Anwenderprogramm anlegen, können Sie das Technologieobjekt später beim Einfügen der PID-Anweisung auswählen.</p>	
<p>Um ein Technologieobjekt auszuwählen, doppelklicken Sie in der Projektnavigation auf das Symbol "Neues Objekt hinzufügen".</p>	

Klicken Sie auf das Symbol "Regeln" und wählen Sie das Technologieobjekt für die Art des PID-Reglers aus (PID\_Compact oder PID\_3Step). Sie können wahlweise auch einen Namen für das Technologieobjekt eingeben.

Klicken Sie auf "OK", um das Technologieobjekt anzulegen.



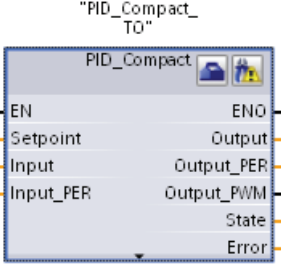
### 9.2.2 Operation PID\_Compact

Der PID-Regler berechnet den Ausgangswert für die Anweisung PID\_Compact anhand folgender Formel.

$$y = K_p \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	Ausgangswert	x	Prozesswert
w	Sollwert	s	Laplace-Operator
K <sub>p</sub>	Proportionale Verstärkung (P-Anteil)	a	Koeffizient für den Differenziervorlauf (D-Anteil)
T <sub>i</sub>	Integralzeit (I-Anteil)	b	Gewichtung des Proportionalanteils (P-Anteil)
T <sub>D</sub>	Differentialzeit (D-Anteil)	c	Gewichtung des Differentialanteils (D-Anteil)

Tabelle 9- 13 Anweisung PID\_Compact

KOP/FUP	SCL	Beschreibung
	<pre>"PID_Compact_1" (     Setpoint:= _real_in_,     Input:= _real_in_,     Input_PER:= _word_in_,     ManualEnable:= _bool_in_,     ManualValue:= _real_in_,     Reset:= _bool_in_,     ScaledInput=&gt; _real_out_,     Output=&gt; _real_out_,     Output_PER=&gt; _word_out_,     Output_PWM=&gt; _bool_out_,     SetpointLimit_H=&gt; _bool_out_,     SetpointLimit_L=&gt; _bool_out_,     InputWarning_H=&gt; _bool_out_,     InputWarning_L=&gt; _bool_out_,     State=&gt; _int_out_,     Error=&gt; dword out );</pre>	<p>PID_Compact bietet einen PID-Regler mit Selbsteinstellung für den Automatik- und Handbetrieb. PID_Compact ist ein PIDT1-Regler mit Anti-Windup und Gewichtung des P- und D-Anteils.</p>

- STEP 7 erstellt das Technologieobjekt und den Instanz-DB automatisch, wenn Sie die Anweisung einfügen. Der Instanz-DB enthält die Parameter des Technologieobjekts.
- Im SCL-Beispiel ist "PID\_Compact\_1" der Name des Instanz-DBs.

Tabelle 9- 14 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
Setpoint	IN	Real	Sollwert des PID-Reglers im Automatikbetrieb. Standardwert: 0.0
Input	IN	Real	Prozesswert. Standardwert: 0.0 Sie müssen außerdem sPid_Cmpt.b_Input_PER_On = FALSCH setzen.
Input_PER	IN	Word	Analoger Prozesswert (optional). Standardwert: W#16#0 Sie müssen außerdem sPid_Cmpt.b_Input_PER_On = WAHR setzen.
ManualEnable	IN	Bool	Aktiviert oder deaktiviert den Handbetrieb. Standardwert: FALSE: <ul style="list-style-type: none"> <li>PID_Compact V1.0 und V1.2: Ist ManualEnable = TRUE, wenn die CPU in den Betriebszustand RUN wechselt, startet PID_Compact im Handbetrieb. Es ist kein Wechsel von FALSE nach TRUE erforderlich, um PID_Compact in den Handbetrieb zu versetzen.</li> <li>PID_Compact V1.1: Ist ManualEnable = TRUE, wenn die CPU in den Betriebszustand RUN wechselt, startet PID Compact im letzten Zustand. Es ist ein Wechsel von TRUE nach FALSE nach TRUE erforderlich, um PID_Compact in den Handbetrieb zu versetzen.</li> </ul>
ManualValue	IN	Real	Prozesswert für Handbetrieb. Standardwert: 0.0
Reset	IN	Bool	Der Parameter Reset startet den Regler neu. Standardwert: FALSCH Im Abschnitt "Reaktion auf Rücksetzen" unten finden Sie Abbildungen der Rücksetzreaktion von <b>PID_Compact</b> V1.1 und V1.0.
ScaledInput	OUT	Real	Skalierter Prozesswert. Standardwert: 0.0
Output <sup>1</sup>	OUT	Real	Ausgangswert. Standardwert: 0.0

Parameter und Datentyp	Datentyp	Beschreibung	
Output_PER <sup>1</sup>	OUT	Word	Analoger Ausgangswert. Standardwert: W#16#0
Output_PWM <sup>1</sup>	OUT	Bool	Ausgangswert für die Impulsdauermodulation. Standardwert: FALSCH
SetpointLimit_H	OUT	Bool	Sollwert oberer Grenzwert. Standardwert: FALSCH Wenn SetpointLimit_H = WAHR, ist der absolute obere Grenzwert des Sollwerts erreicht. Standardwert: FALSCH
SetpointLimit_L	OUT	Bool	Sollwert unterer Grenzwert. Standardwert: FALSCH Wenn SetpointLimit_L = WAHR, ist der absolute untere Grenzwert des Sollwerts erreicht. Standardwert: FALSCH
InputWarning_H	OUT	Bool	Wenn InputWarning_H = WAHR, hat der Prozesswert die obere Warngrenze erreicht oder überschritten. Standardwert: FALSCH
InputWarning_L	OUT	Bool	Wenn InputWarning_L = WAHR, hat der Prozesswert die untere Warngrenze erreicht. Standardwert: FALSCH
State	OUT	Int	Aktuelle Betriebsart des PID-Reglers. Standardwert: 0 Mit sRet.i_Mode wechseln Sie die Betriebsart. <ul style="list-style-type: none"> <li>• State = 0: Inaktiv</li> <li>• State = 1: Erstopptimierung</li> <li>• State = 2: Manuelle Feineinstellung</li> <li>• State = 3: Automatikbetrieb</li> <li>• State = 4: Handbetrieb</li> </ul>
ErrorBits	OUT	DWord	In der Parametertabelle (Seite 375) der ErrorBits der Anweisung PID_Compact werden die Fehlermeldungen beschrieben. Standardwert: DW#16#0000 (kein Fehler)

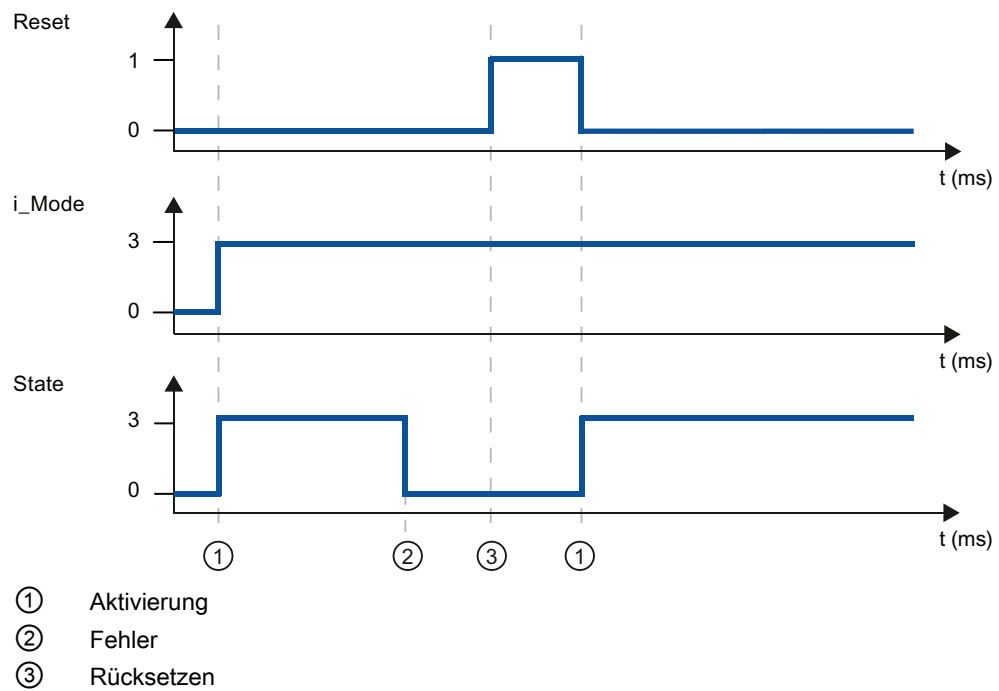
<sup>1</sup> Die Ausgänge der Parameter Output, Output\_PER und Output\_PWM können parallel verwendet werden.

### Reaktion auf Rücksetzen

Die Reaktion auf Rücksetzen = TRUE richtet sich nach der Version der Anweisung PID\_Compact.

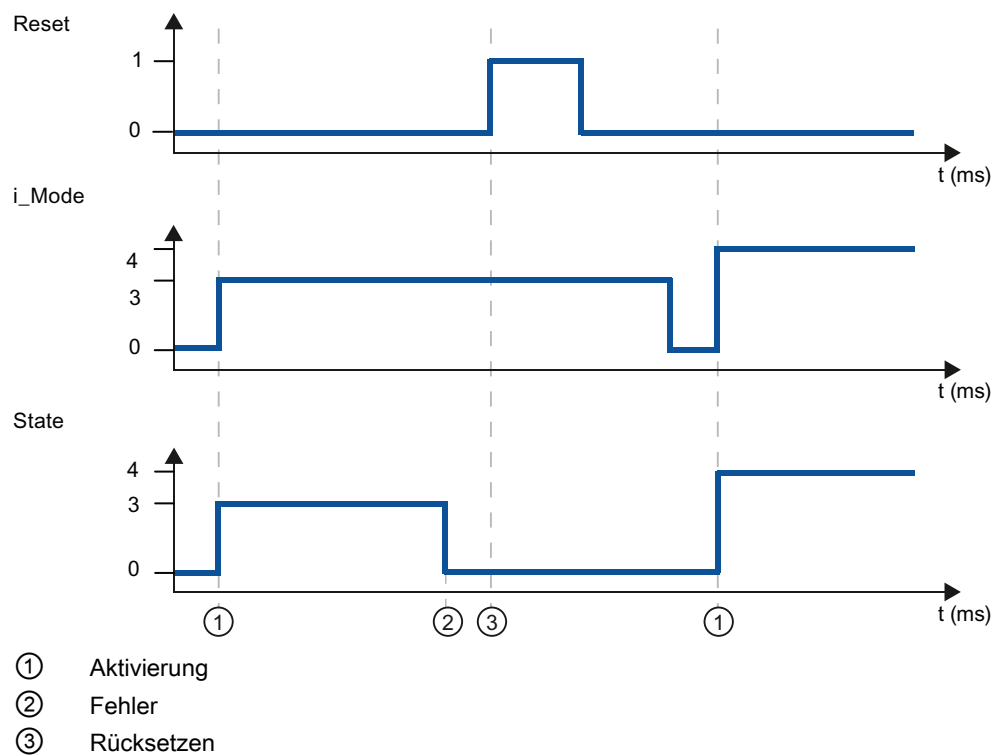
#### Rücksetzreaktion von PID\_Compact V1.1

Eine steigende Flanke am Eingang Reset setzt die Fehler und Warnungen zurück und löscht die integrierte Aktion. Eine fallende Flanke am Eingang Reset bewirkt einen Wechsel in den zuletzt aktiven Betriebszustand.



#### Rücksetzreaktion von PID\_Compact V1.0

Eine steigende Flanke am Eingang Reset setzt die Fehler und Warnungen zurück und löscht die integrierte Aktion. Der Regler wird erst bei der nächsten Flanke an i\_Mode reaktiviert.



Funktionsweise des Reglers PID\_Compact

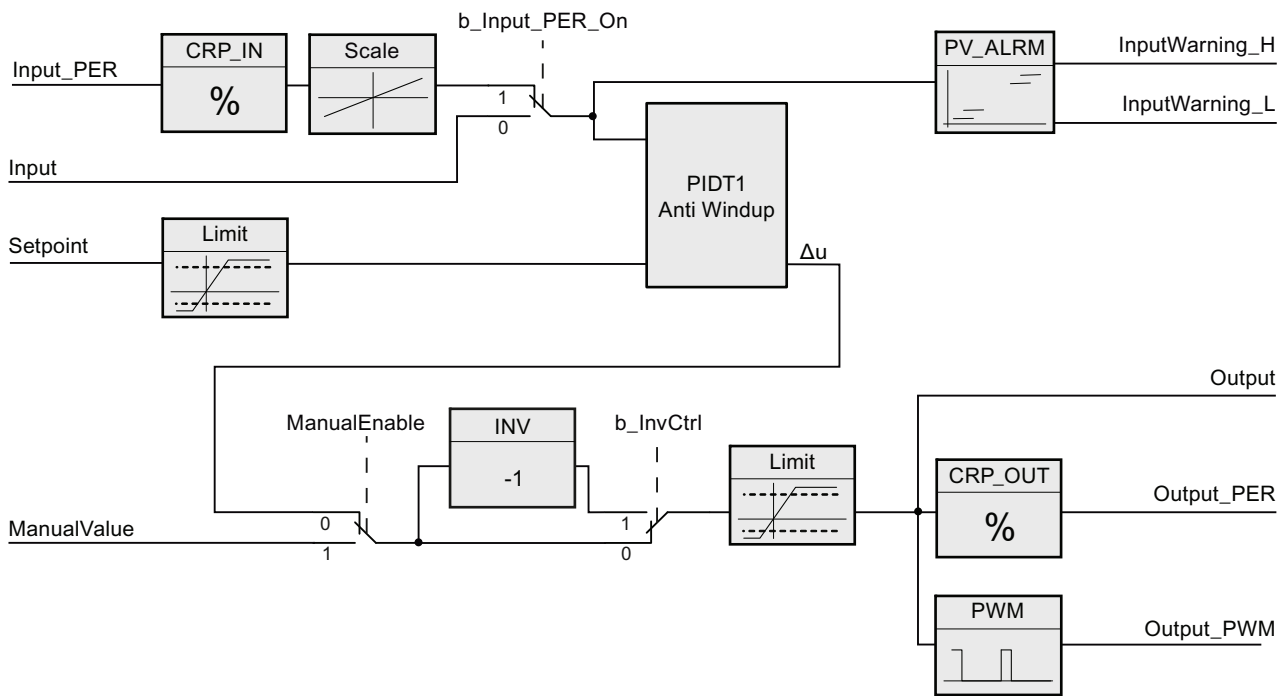


Bild 9-1 Funktionsweise des Reglers PID\_Compact

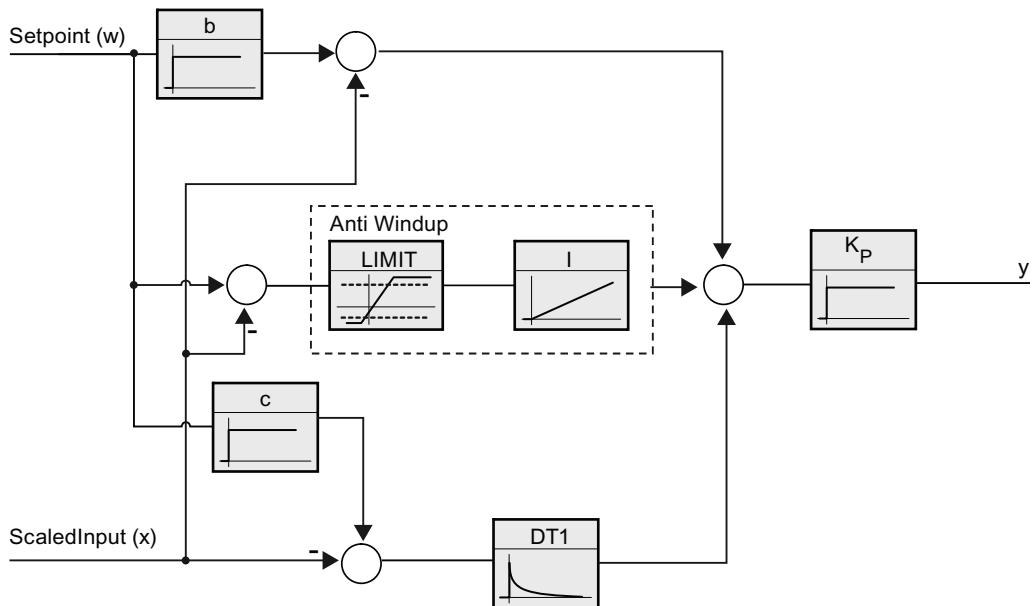


Bild 9-2 Funktionsweise des Reglers PID\_Compact als PIDE1-Regler mit Anti-Windup

### 9.2.3 ErrorBit-Parameter der Anweisung PID\_Compact

Stehen mehrere Fehler an, werden die Werte der Fehlercodes mittels binärer Addition angezeigt. Die Anzeige von Fehlercode 0003 beispielsweise weist darauf hin, dass auch die Fehler 0001 und 0002 anstehen.

Tabelle 9- 15 ErrorBit-Parameter der Anweisung PID\_Compact

ErrorBit (DW#16#...)	Beschreibung
0000	Kein Fehler
0001	Der Parameter "Input" liegt außerhalb der Prozesswertgrenzen. Input > sPid_Cmpt.r_Pv_Hlmo Input < sPid_Cmpt.r_Pv_Llm Sie können den Aktor erst dann wieder starten, wenn der Fehler behoben ist.
0002	Ungültiger Wert am Parameter "Input_PER". Prüfen Sie, ob am Analogeingang ein Fehler anliegt.
0004	Fehler bei der Feineinstellung. Die Schwingung des Prozesswerts konnte nicht erhalten werden.
0008	Fehler beim Start der Voreinstellung. Der Prozesswert ist zu nah am Sollwert. Starten Sie die Feineinstellung.
0010	Der Sollwert wurde während der Einstellung des Reglers geändert.
0020	Die Voreinstellung kann nicht im Automatikbetrieb oder während der Feineinstellung durchgeführt werden.
0040	Fehler bei der Feineinstellung. Der Sollwert ist zu nah an den Grenzen des Sollwerts.
0080	Inkorrekte Konfiguration der Grenzen des Ausgangswerts. Prüfen Sie, ob die Grenzen des Ausgangswerts richtig konfiguriert sind, und passen Sie die Richtung an, in die der Regler arbeitet.
0100	Fehler während der Reglereinstellung führte zu ungültigen Parametern.
0200	Ungültiger Wert am Parameter "Input": Das numerische Format des Werts ist ungültig.
0400	Berechnung des Ausgangswerts schlug fehl. Prüfen Sie die PID-Parameter.
0800	Abtastzeitfehler: PID_Compact wird nicht während der Abtastzeit des Weckalarm-OBs aufgerufen.
1000	Ungültiger Wert am Parameter "Setpoint": Das numerische Format des Werts ist ungültig.

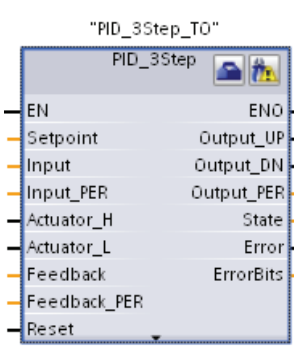
### 9.2.4 Anweisung PID\_3STEP

Der PID-Regler berechnet den Ausgangswert für die Anweisung PID\_3Step anhand folgender Formel.

$$\Delta y = K_p \cdot s \cdot \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	Ausgangswert	x	Prozesswert
w	Sollwert	s	Laplace-Operator
K <sub>p</sub>	Proportionale Verstärkung (P-Anteil)	a	Koeffizient für den Differenziervorlauf (D-Anteil)
T <sub>i</sub>	Integralzeit (I-Anteil)	b	Gewichtung des Proportionalanteils (P-Anteil)
T <sub>D</sub>	Differentialzeit (D-Anteil)	c	Gewichtung des Differentialanteils (D-Anteil)

Tabelle 9- 16 Anweisung PID\_3Step

KOP/FUP	SCL	Beschreibung
	<pre>"PID_3Step_1" (     SetpoInt:=_real_in_,     Input:=_real_in_,     ManualValue:=_real_in_,     Feedback:=_real_in_,     InputPer:=_word_in_,     FeedbackPer:=_word_in_,     ManualEnable:=_bool_in_,     ManualUP:=_bool_in_,     ManualDN:=_bool_in_,     ActuatorH:=_bool_in_,     ActuatorL:=_bool_in_,     Reset:=_bool_in_,     ScaledInput=&gt;_real_out_,     ScaledFeedback=&gt;_real_out_,     ErrorBits=&gt;_dword_out_,     OutputPer=&gt;_word_out_,     State=&gt;_int_out_,     OutputUP=&gt;_bool_out_,     OutputDN=&gt;_bool_out_,     SetpoIntLimitH=&gt;_bool_out_,     SetpoIntLimitL=&gt;_bool_out_,     InputWarningH=&gt;_bool_out_,     InputWarningL=&gt;_bool_out_,     Error=&gt;_bool_out_);</pre>	<p>PID_3Step konfiguriert einen PID-Regler mit Selbsteinstellungsfunktionen, der für motorbetätigte Ventile und Stellglieder optimiert wurde. Er bietet zwei Boolesche Ausgänge.</p> <p>PID_3Step ist ein PIDT1-Regler mit Anti-Windup und Gewichtung des P- und D-Anteils.</p>

- STEP 7 erstellt das Technologieobjekt und den Instanz-DB automatisch, wenn Sie die Anweisung einfügen. Der Instanz-DB enthält die Parameter des Technologieobjekts.
- Im SCL-Beispiel ist "PID\_3Step\_1" der Name des Instanz-DBs.



Tabelle 9- 17 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Setpoint	IN	Real	Sollwert des PID-Reglers im Automatikbetrieb. Standardwert: 0.0
Input	IN	Real	Prozesswert. Standardwert: 0.0 Sie müssen außerdem Config.InputPEROn = FALSE setzen.
Input_PER	IN	Word	Analoger Prozesswert (optional). Standardwert: W#16#0 Sie müssen außerdem Config.InputPEROn = TRUE setzen.
ManualEnable	IN	Bool	Aktiviert oder deaktiviert den Handbetrieb. Standardwert: FALSE <ul style="list-style-type: none"> <li>Bei der Flanke von FALSE nach TRUE wechselt der PID-Regler in den Handbetrieb, State = 4 und Retain.Mode ändern sich nicht.</li> <li>Bei der Flanke von TRUE nach FALSE wechselt der PID-Regler in die letzte aktive Betriebsart und nach State = Retain.Mode.</li> </ul>
ManualUP	IN	Bool	Im Handbetrieb wird das Ventil durch jede steigende Flanke um 5% des Gesamtstellbereichs geöffnet, bzw. für die Dauer der minimalen Motorbetätigungszeit. ManualUP wird nur ausgewertet, wenn Sie Output_PER nicht verwenden <b>und</b> es keine Positionsrückmeldung gibt. Standardwert: FALSE <ul style="list-style-type: none"> <li>Wenn Output_PER FALSE ist, schaltet der manuelle Eingang Output_UP für den Zeitraum ein, der einer Bewegung von 5 % des Geräts entspricht.</li> <li>Wenn Config.ActuatorEndStopOn TRUE ist, dann wird Output_UP nicht eingeschaltet, wenn Actuator_H TRUE ist.</li> </ul>
ManualDN	IN	Bool	Im Handbetrieb wird das Ventil durch jede steigende Flanke um 5 % des Gesamtstellbereichs geschlossen, bzw. für die Dauer der minimalen Motorbetätigungszeit. ManualDN wird nur ausgewertet, wenn Sie Output_PER nicht verwenden <b>und</b> es keine Positionsrückmeldung gibt. Standardwert: FALSE <ul style="list-style-type: none"> <li>Wenn Output_PER FALSE ist, schaltet der manuelle Eingang Output_DN für den Zeitraum ein, der einer Bewegung von 5 % des Geräts entspricht.</li> <li>Wenn Config.ActuatorEndStopOn TRUE ist, dann wird Output_DN nicht eingeschaltet, wenn Actuator_L TRUE ist.</li> </ul>
ManualValue	IN	Real	Prozesswert für Handbetrieb. Standardwert: 0.0 Im Handbetrieb geben Sie die absolute Position des Ventils an. ManualValue wird nur ausgewertet, wenn Sie OutputPer verwenden <b>oder</b> wenn eine zurückgemeldete Position verfügbar ist. Standardwert: 0.0
Feedback	IN	Real	Zurückgemeldete Position des Ventils. Standardwert: 0.0 Um Feedback zu verwenden, setzen Sie Config.FeedbackPerOn = FALSE.

Parameter und Datentyp		Datentyp	Beschreibung
Feedback_PER	IN	Word	Zurückgemeldeter Analogwert der Ventilposition. Standardwert: W#16#0 Um Feedback_PER zu verwenden, setzen Sie Config.FeedbackPerOn = TRUE. Feedback_PER wird mit folgenden Parametern skaliert: <ul style="list-style-type: none"> <li>• Config.FeedbackScaling.LowerPointIn</li> <li>• Config.FeedbackScaling.UpperPointIn</li> <li>• Config.FeedbackScaling.LowerPointOut</li> <li>• Config.FeedbackScaling.UpperPointOut</li> </ul>
Actuator_H	IN	Bool	Wenn Actuator_H = TRUE, befindet sich das Ventil am oberen Endpunkt und wird nicht weiter in diese Richtung bewegt. Standardwert: FALSE
Actuator_L	IN	Bool	Wenn Actuator_L = TRUE, befindet sich das Ventil am unteren Endpunkt und wird nicht weiter in diese Richtung bewegt. Standardwert: FALSE
Reset	IN	Bool	Startet den PID-Regler neu. Standardwert: FALSE Bei einer Flanke von FALSE nach TRUE: <ul style="list-style-type: none"> <li>• Betriebszustand "Inaktiv"</li> <li>• Eingangswert = 0</li> <li>• Zwischenwerte des Reglers werden zurückgesetzt. (PID-Parameter werden gespeichert.)</li> </ul> Bei einer Flanke von TRUE nach FALSE wird in den zuletzt aktiven Betriebszustand gewechselt.
ScaledInput	OUT	Real	Skalierter Prozesswert
ScaledFeedback	OUT	Real	Skalierte Ventilposition
Output_PER	OUT	Word	Analoger Ausgangswert. Wenn Config.OutputPerOn = TRUE, dann wird der Parameter Output_PER verwendet.
Output_UP	OUT	Bool	Digitaler Ausgangswert zum Öffnen des Ventils. Standardwert: FALSE Wenn Config.OutputPerOn = FALSE, dann wird der Parameter Output_UP verwendet.
Output_DN	OUT	Bool	Digitaler Ausgangswert zum Schließen des Ventils. Standardwert: FALSE Wenn Config.OutputPerOn = FALSE, dann wird der Parameter Output_DN verwendet.
SetpointLimitH	OUT	Bool	Sollwert oberer Grenzwert. Standardwert: FALSE Wenn SetpointLimitH = TRUE, dann ist der absolute obere Grenzwert des Sollwerts erreicht. In der CPU ist der Sollwert auf den konfigurierten absoluten oberen Grenzwert des Istwerts begrenzt.
SetpointLimitL	OUT	Bool	Sollwert unterer Grenzwert. Standardwert: FALSE Wenn SetpointLimitL = TRUE, dann ist der absolute untere Grenzwert des Sollwerts erreicht. In der CPU ist der Sollwert auf den konfigurierten absoluten unteren Grenzwert des Istwerts begrenzt.
InputWarningH	OUT	Bool	Wenn InputWarningH = TRUE, hat der Eingangswert die obere Warngrenze erreicht oder überschritten. Standardwert: FALSE
InputWarningL	OUT	Bool	Wenn InputWarningL = TRUE, hat der Eingangswert die untere Warngrenze erreicht oder überschritten. Standardwert: FALSE

Parameter und Datentyp	Datentyp	Beschreibung
State	OUT	Int Aktuelle Betriebsart des PID-Reglers. Standardwert: 0 Mit Retain.Mode wechseln Sie die Betriebsart: <ul style="list-style-type: none"> <li>• State = 0: Inaktiv</li> <li>• State = 1: Ersoptimierung</li> <li>• State = 2: Manuelle Feineinstellung</li> <li>• State = 3: Automatikbetrieb</li> <li>• State = 4: Handbetrieb</li> <li>• State = 5: Ersatzausgangswert</li> <li>• State = 6: Übergangszeitmessung</li> <li>• State = 7: Ersatzausgangswert mit Fehlerüberwachung</li> <li>• State = 8: Fehlerüberwachung</li> </ul>
Error	OUT	Bool Wenn Error = TRUE, steht mindestens eine Fehlermeldung an. Standardwert: FALSE
ErrorBits	OUT	DWord In der Parametertabelle (Seite 382) der ErrorBits der Anweisung PID_3STEP werden die Fehlermeldungen beschrieben. Standardwert: DW#16#0000 (kein Fehler)

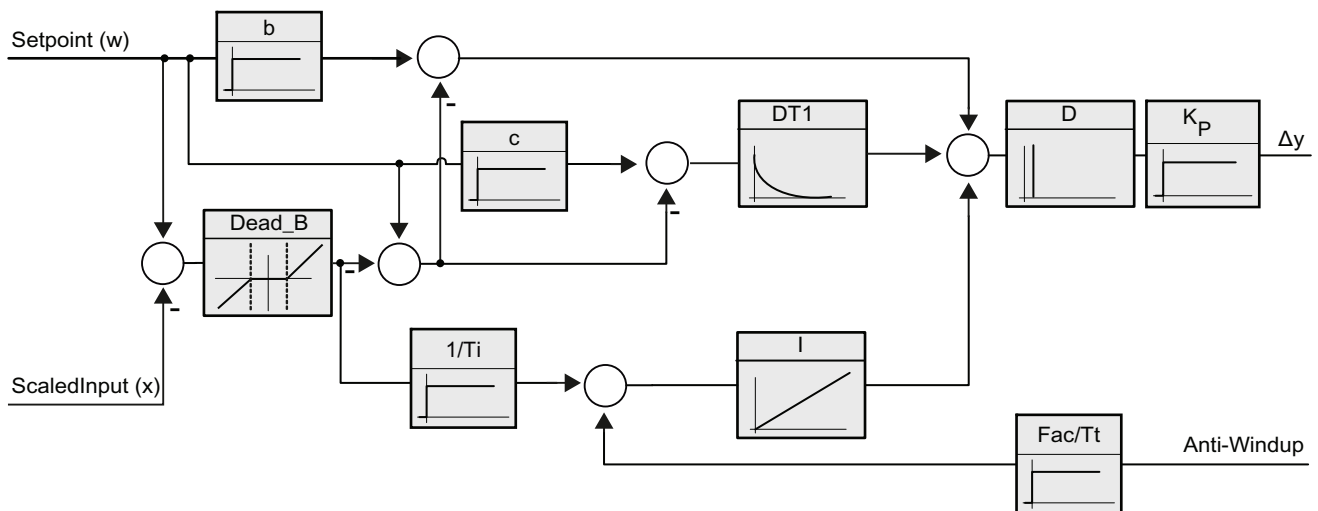


Bild 9-3 Funktionsweise des Reglers PID\_3Step als PIDT1-Regler mit Anti-Windup

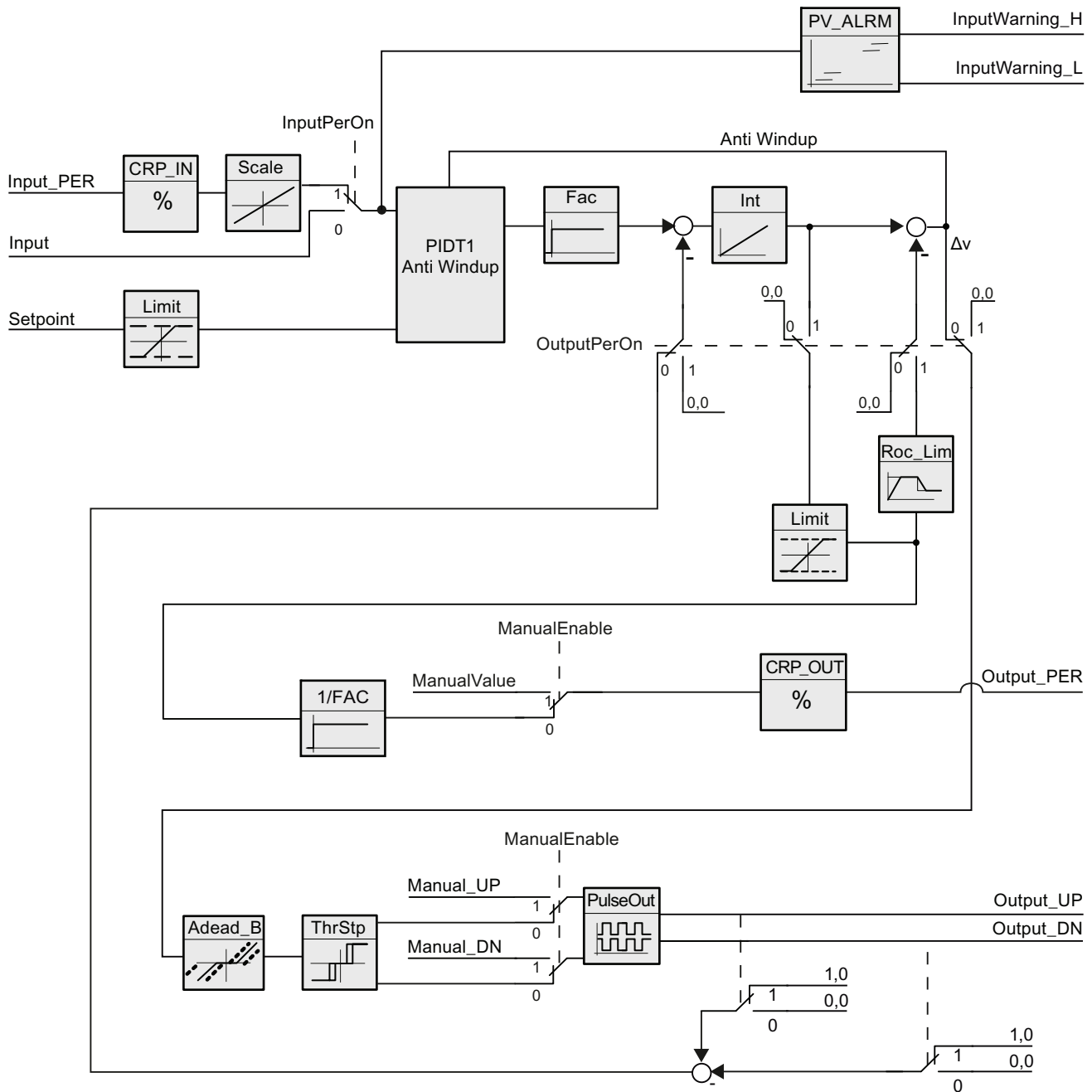


Bild 9-4 Funktionsweise des Reglers PID\_3Step ohne Positionsrückmeldung

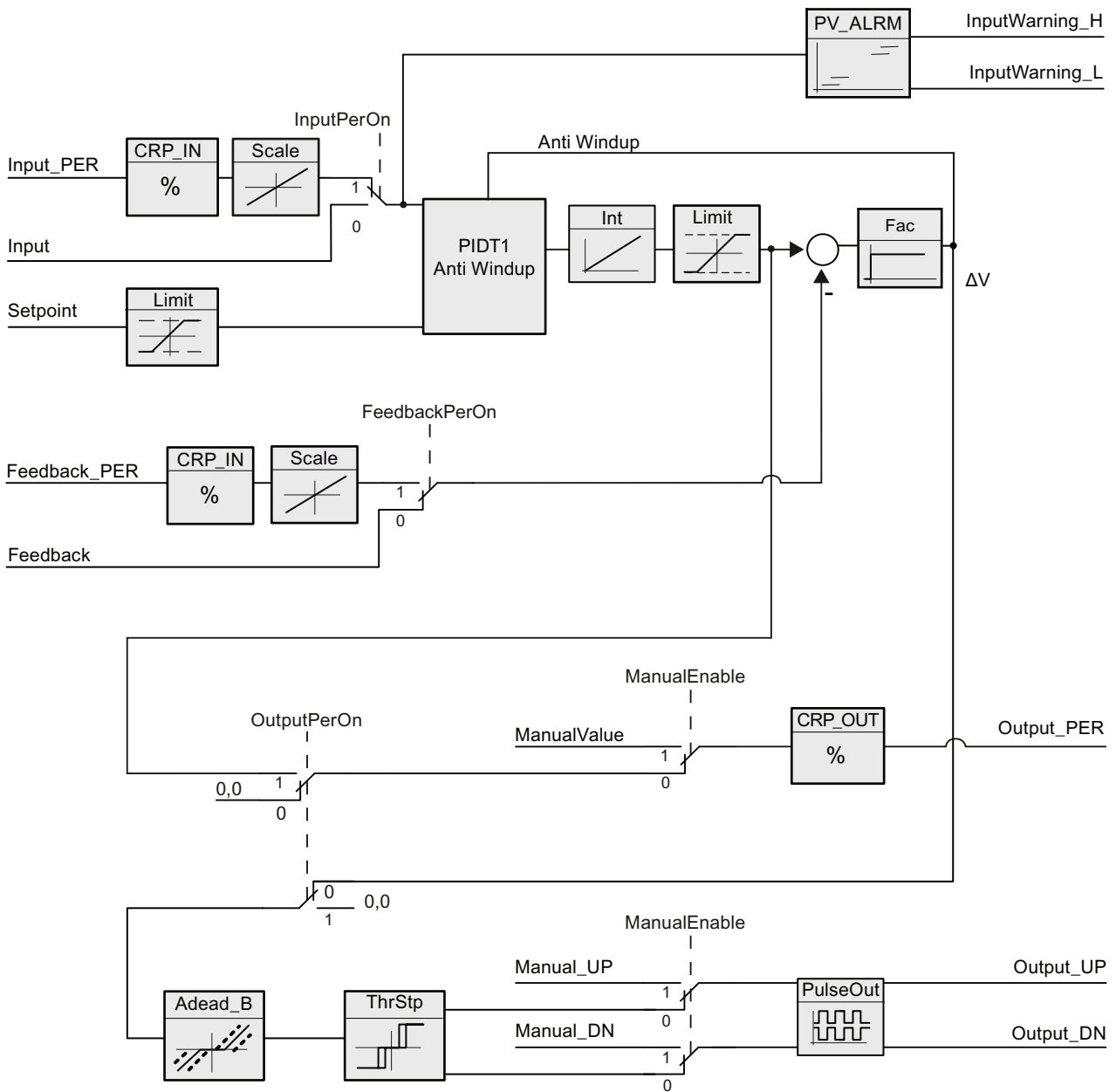


Bild 9-5 Funktionsweise des Reglers PID\_3Step mit aktivierter Positionsrückmeldung

### 9.2.5 ErrorBit-Parameter der Anweisung PID\_3STEP

Stehen mehrere Fehler an, werden die Werte der Fehlercodes mittels binärer Addition angezeigt. Die Anzeige von Fehlercode 0003 beispielsweise weist darauf hin, dass auch die Fehler 0001 und 0002 anstehen.

Tabelle 9- 18 ErrorBit-Parameter der Anweisung PID\_3STEP

ErrorBit (DW#16#...)	Beschreibung
0000	Kein Fehler
0001	Der Parameter "Input" liegt außerhalb der Prozesswertgrenzen: Input > Config.InputUpperLimit oder Input < Config.InputLowerLimit Wenn ActivateRecoverMode = TRUE und ErrorBehaviour = 1, wechselt der Aktor zum Ersatzausgangswert. Wenn ActivateRecoverMode = TRUE und ErrorBehaviour = 0, stoppt der Aktor an seiner aktuellen Position. Wenn ActivateRecoverMode = FALSE, stoppt der Aktor an seiner aktuellen Position. PID_3STEP V1.1: Sie können den Aktor in den Handbetrieb versetzen. PID_3STEP V1.0: In diesem Zustand ist kein Handbetrieb möglich. Sie können den Aktor erst dann wieder starten, wenn der Fehler behoben ist.
0002	Ungültiger Wert am Parameter "Input_PER". Prüfen Sie, ob am Analogeingang ein Fehler anliegt. Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ist ActivateRecoverMode = TRUE und der Fehler steht nicht mehr an, PID_3STEP geht wieder in den Automatikbetrieb.
0004	Fehler bei der Feineinstellung. Die Schwingung des Prozesswerts konnte nicht erhalten werden.
0008	Fehler beim Start der Voreinstellung. Der Prozesswert ist zu nah am Sollwert. Starten Sie die Feineinstellung.
0010	Der Sollwert darf während der Feineinstellung nicht geändert werden.
0020	Die Voreinstellung kann nicht im Automatikbetrieb oder während der Feineinstellung durchgeführt werden.
0040	Fehler bei der Feineinstellung. Der Sollwert ist zu nah an den Grenzen des Sollwerts.
0080	Fehler bei der Voreinstellung. Inkorrekte Konfiguration der Grenzen des Ausgangswerts. Prüfen Sie, ob die Grenzen des Ausgangswerts richtig konfiguriert sind, und passen Sie die Richtung an, in die der Regler arbeitet.
0100	Fehler während der Feineinstellung führte zu ungültigen Parametern.
0200	Ungültiger Wert am Parameter "Input": Das numerische Format des Werts ist ungültig. Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ist ActivateRecoverMode = TRUE und der Fehler steht nicht mehr an, PID_3STEP geht wieder in den Automatikbetrieb.
0400	Berechnung des Ausgangswerts schlug fehl. Prüfen Sie die PID-Parameter.

ErrorBit (DW#16#...)	Beschreibung
0800	<p>Abtastzeitfehler: PID_3STEP wird nicht während der Abtastzeit des Weckalarm-OBs aufgerufen.</p> <p>Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ist ActivateRecoverMode = TRUE und der Fehler steht nicht mehr an, PID_3STEP geht wieder in den Automatikbetrieb.</p>
1000	<p>Ungültiger Wert am Parameter "Setpoint": Das numerische Format des Werts ist ungültig.</p> <p>Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ist ActivateRecoverMode = TRUE und der Fehler steht nicht mehr an, PID_3STEP geht wieder in den Automatikbetrieb.</p>
2000	<p>Ungültiger Wert am Parameter Feedback_PER.</p> <p>Prüfen Sie, ob am Analogeingang ein Fehler anliegt.</p> <p>Der Aktor kann nicht zum Ersatzausgangswert wechseln und verlässt die aktuelle Position nicht. In diesem Zustand ist kein Handbetrieb möglich. Sie müssen die Positionsrückmeldung deaktivieren (Config.FeedbackOn = FALSE), um den Aktor aus diesem Zustand herauszuholen.</p> <p>Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ist ActivateRecoverMode = TRUE und der Fehler steht nicht mehr an, PID_3STEP geht wieder in den Automatikbetrieb.</p>
4000	<p>Ungültiger Wert am Parameter Feedback. Das numerische Format des Werts ist ungültig.</p> <p>Der Aktor kann nicht zum Ersatzausgangswert wechseln und verlässt die aktuelle Position nicht. In diesem Zustand ist kein Handbetrieb möglich. Sie müssen die Positionsrückmeldung deaktivieren (Config.FeedbackOn = FALSE), um den Aktor aus diesem Zustand herauszuholen.</p> <p>Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ist ActivateRecoverMode = TRUE und der Fehler steht nicht mehr an, PID_3STEP geht wieder in den Automatikbetrieb.</p>
8000	<p>Fehler bei der digitalen Positionsrückmeldung. Actuator_H = TRUE und Actuator_L = TRUE.</p> <p>Der Aktor kann nicht zum Ersatzausgangswert wechseln und verlässt die aktuelle Position nicht. In diesem Zustand ist kein Handbetrieb möglich.</p> <p>Sie müssen "Endstopp signalisiert Aktor" deaktivieren (Config.ActuatorEndStopOn = FALSE), um den Aktor aus diesem Zustand herauszuholen.</p> <p>Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ist ActivateRecoverMode = TRUE und der Fehler steht nicht mehr an, PID_3STEP geht wieder in den Automatikbetrieb.</p>

### 9.2.6 PID-Regler konfigurieren

Die Parameter des Technologieobjekts legen die Funktionsweise des PID-Reglers fest. Öffnen Sie den Konfigurationseditor über das Symbol.

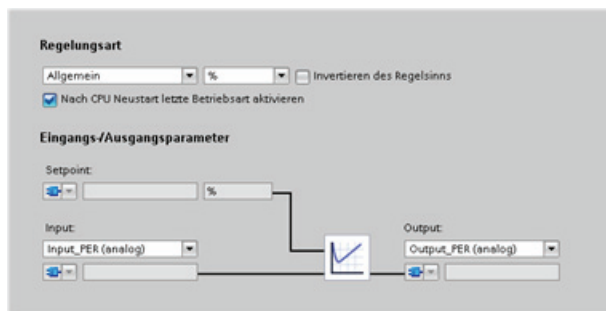


Bild 9-6 Konfigurationseditor für PID\_Compact (Grundlegende Einstellungen)

Tabelle 9- 19 Beispielhafte Konfigurationseinstellungen für die Anweisung PID\_Compact

Einstellungen		Beschreibung
Grundlagen	Reglertyp	Wählt die physikalischen Einheiten aus.
	Reglerlogik invertieren	<p>Ermöglicht die Auswahl eines invers funktionierenden PID-Reglers.</p> <ul style="list-style-type: none"> <li>Ist diese Option nicht ausgewählt, verhält sich der PID-Regler entsprechend der direkten Funktionsweise und der Ausgang des PID-Regelkreises steigt, wenn der Eingangswert &lt; Sollwert ist.</li> <li>Ist diese Option ausgewählt, steigt der Ausgang des PID-Regelkreises, wenn der Eingangswert &gt; Sollwert ist.</li> </ul>
	Nach CPU-Neustart letzte Betriebsart aktivieren	Startet den PID-Regler neu, nachdem er zurückgesetzt wurde oder wenn eine Eingangsgrenze überschritten und in den gültigen Bereich zurückgeführt wurde.
	Eingang	Wählt entweder den Parameter Input oder den Parameter Input_PER (bei analog) für den Prozesswert aus. Input_PER kann direkt von einem analogen Eingangsmodul kommen.
	Ausgang	Wählt entweder den Parameter Output oder den Parameter Output_PER (bei analog) für den Ausgangswert aus. Output_PER kann direkt an ein analoges Ausgangsmodul gehen.
Prozesswert	<p>Skaliert sowohl den Bereich als auch die Grenzen für den Prozesswert. Wenn der Prozesswert unter die untere Grenze fällt oder über die obere Grenze steigt, wechselt der PID-Regelkreis in die inaktive Betriebsart und setzt den Ausgangswert auf 0.</p> <p>Um Input_PER zu verwenden, <b>müssen</b> Sie den analogen Prozesswert (Eingangswert) skalieren.</p>	



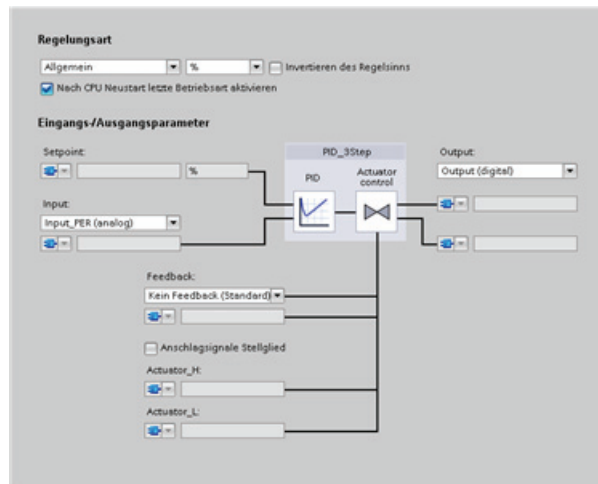


Bild 9-7 Konfigurationseditor für PID\_3Step (Grundlegende Einstellungen)

Tabelle 9- 20 Beispielhafte Konfigurationseinstellungen für die Anweisung PID\_3Step

Einstellungen		Beschreibung
Grundlagen	Reglertyp	Wählt die physikalischen Einheiten aus.
	Reglerlogik invertieren	Ermöglicht die Auswahl eines invers funktionierenden PID-Reglers. <ul style="list-style-type: none"> <li>Ist diese Option nicht ausgewählt, verhält sich der PID-Regler entsprechend der direkten Funktionsweise und der Ausgang des PID-Regelkreises steigt, wenn der Eingangswert &lt; Sollwert ist.</li> <li>Ist diese Option ausgewählt, steigt der Ausgang des PID-Regelkreises, wenn der Eingangswert &gt; Sollwert ist.</li> </ul>
	Nach CPU-Neustart letzte Betriebsart aktivieren	Startet den PID-Regler neu, nachdem er zurückgesetzt wurde oder wenn eine Eingangsgrenze überschritten und in den gültigen Bereich zurückgeführt wurde.
	Eingang	Wählt entweder den Parameter Input oder den Parameter Input_PER (bei analog) für den Prozesswert aus. Input_PER kann direkt von einem analogen Eingangsmodul kommen.
	Ausgang	Gibt an, ob die digitalen Ausgänge (Output_UP und Output_DN) oder der analoge Ausgang (Output_PER) für den Ausgangswert verwendet werden soll.
	Rückmeldung	Gibt die Art des an den PID-Regler zurückgemeldeten Gerätezustands an: <ul style="list-style-type: none"> <li>Keine Rückmeldung (Standard)</li> <li>Rückmeldung</li> <li>Feedback_PER</li> </ul>
Prozesswert	Skaliert sowohl den Bereich als auch die Grenzen für den Prozesswert. Wenn der Prozesswert unter die untere Grenze fällt oder über die obere Grenze steigt, wechselt der PID-Regelkreis in die inaktive Betriebsart und setzt den Ausgangswert auf 0. Um Input_PER zu verwenden, <b>müssen</b> Sie den analogen Prozesswert (Eingangswert) skalieren.	
Stellglied	Motorübergangszeit	Legt die Zeit vom offenen zum geschlossenen Ventil fest. (Ermitteln Sie diesen Wert auf dem Datenblatt oder anhand des Ventil-Typenschildes.)
	Kleinste EIN-Zeit	Legt die minimale Betätigungszeit für das Ventil fest. (Ermitteln Sie diesen Wert auf dem Datenblatt oder anhand des Ventil-Typenschildes.)

Einstellungen		Beschreibung
	Kleinste AUS-Zeit	Legt die minimale Pausenzeit für das Ventil fest. (Ermitteln Sie diesen Wert auf dem Datenblatt oder anhand des Ventil-Typenschildes.)
	Fehlerverhalten	Definiert das Verhalten des Ventils, wenn ein Fehler erkannt oder der PID-Regler zurückgesetzt wird. Wenn Sie festlegen, dass eine Ersatzposition verwendet werden soll, geben Sie die "Sicherheitsposition" ein. Bei analoger Rückmeldung oder analogem Ausgang wählen Sie einen Wert zwischen der unteren und oberen Grenze für den Ausgang. Bei Digitalausgängen können Sie nur 0 % (aus) oder 100 % (ein) wählen.
	Positionsrückmeldung skalieren <sup>1</sup>	<ul style="list-style-type: none"> <li>• "Stopp oben" und "Stopp untere Grenze" definieren die maximale positive Position (vollständig geöffnet) und die maximale negative Position (vollständig geschlossen). "Stopp oben" muss größer als "Stopp untere Grenze" sein.</li> <li>• "Prozesswert obere Grenze" und "Prozesswert untere Grenze" definieren die untere und obere Position des Ventils während der Einstellung und im Automatikbetrieb.</li> <li>• "FeedbackPER" ("Low" und "High") definieren die analoge Rückmeldung der Ventilposition. "FeedbackPER High" muss höher als "FeedbackPER Low" sein.</li> </ul>

<sup>1</sup> "Positionsrückmeldung skalieren" kann nur geändert werden, wenn Sie in den grundlegenden Einstellungen "Rückmeldung" aktiviert haben.

### 9.2.7 Inbetriebnahme des PID-Reglers

Sie konfigurieren den PID-Regler im Inbetriebnahme-Editor für die Selbsteinstellung beim Anlauf und für die Selbsteinstellung während des Betriebs. Um den Inbetriebnahme-Editor aufzurufen, klicken Sie im Anweisungsverzeichnis oder in der Projektnavigation auf das entsprechende Symbol.



Tabelle 9- 21 Beispiel für einen Konfigurationsbildschirm (PID\_3Step)

	<ul style="list-style-type: none"> <li>• <b>Messung:</b> Um den Sollwert, den Prozesswert (Eingangswert) und den Ausgangswert in einer Echtzeitkurve anzuzeigen, geben Sie die Abtastzeit ein und klicken auf die Schaltfläche "Start".</li> <li>• <b>Optimierungsart:</b> Um den PID-Regler einzustellen, wählen Sie entweder "Erstoptimierung" oder "Feineinstellung" (manuell) und klicken auf die Schaltfläche "Start". Der PID-Regler durchläuft mehrere Phasen, um die Zeiten für Systemantwort und Aktualisierung zu berechnen. Anhand dieser Werte werden die entsprechenden Optimierungsparameter berechnet.</li> </ul> <p>Nach Abschluss des Einstellungsprozesses können Sie die neuen Parameter speichern, indem Sie im Inbetriebnahme-Editor im Bereich "PID-Parameter" auf die Schaltfläche "PID-Parameter laden" klicken.</p> <p>Tritt während der Einstellung ein Fehler auf, geht der Ausgangswert des PID-Reglers auf 0. Der PID-Regler wird in die "inaktive" Betriebsart versetzt. Der Status zeigt den Fehler an.</p>
--	--

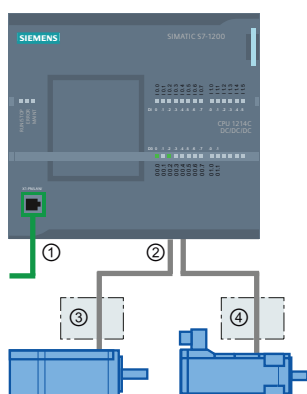
## 9.3 Bewegungssteuerung

Die CPU bietet Bewegungssteuerungsfunktionen für den Betrieb von Schrittmotoren und Servomotoren mit Impulsschnittstelle. Die Bewegungssteuerungsfunktion übernimmt die Steuerung und Überwachung der Antriebe.

- Das Technologieobjekt "Achse" konfiguriert die Daten des mechanischen Antriebs, die Antriebsschnittstelle, die dynamischen Parameter und andere Eigenschaften des Antriebs.
- Sie konfigurieren die Impuls- und Richtungsausträge der CPU zum Steuern des Antriebs.
- Ihr Anwenderprogramm nutzt die Bewegungssteuerungsanweisungen zum Steuern der Achse und zum Auslösen von Bewegungsaufgaben.
- Stellen Sie über die PROFINET-Schnittstelle eine Online-Verbindung zwischen der CPU und dem Programmiergerät her. Neben den Online-Funktionen der CPU stehen für die Bewegungssteuerung weitere Inbetriebnahme- und Diagnosefunktionen zur Verfügung.

### Hinweis

Änderungen, die Sie an der Bewegungssteuerungskonfiguration vornehmen und im Betriebszustand RUN ins Zielsystem laden, werden erst dann wirksam, wenn die CPU von STOP in RUN geht.



- ① PROFINET
- ② Impuls- und Richtungsausgänge
- ③ Leistungsteil Schrittmotor
- ④ Leistungsteil Servomotor

Die DC/DC/DC-Varianten der S7-1200 CPU haben integrierte Ausgänge für die direkte Steuerung von Antrieben. Die Relaisvarianten der CPU benötigen das Signalboard mit DC-Ausgängen für die Antriebssteuerung.

Ein Signalboard (SB) erweitert die integrierten E/A um einige zusätzliche Ein- und Ausgänge. Ein SB mit 2 Digitalausgängen kann als Impuls- und Richtungsausgang zur Steuerung eines Motors verwendet werden. Ein SB mit 4 Digitalausgängen kann für Impuls- und Richtungsausgänge zur Steuerung zweier Motoren verwendet werden. Integrierte Relaisausgänge können nicht als Impulsausgänge zur Motorsteuerung verwendet werden.

**Hinweis**

**Impulsfolgen können von anderen Anweisungen im Anwenderprogramm nicht verwendet werden**

Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM oder Bewegungssteuerungsanweisungen) konfigurieren, werden die entsprechenden Adressen der Ausgänge (A0.0 bis A0.3, A4.0 bis A4.3) aus dem Speicher der Ausgänge entfernt und können in Ihrem Anwenderprogramm nicht für andere Zwecke verwendet werden. Wenn Ihr Anwenderprogramm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.

Tabelle 9- 22 Maximale Anzahl von steuerbaren Antrieben

Typ der CPU		Kein SB installiert	Mit einem SB (2 x DC-Ausgänge)	Mit einem SB (4 x DC-Ausgänge)
CPU 1211C	DC/DC/DC	2	2	2
	AC/DC/RLS	0	1	2
	DC/DC/RLS	0	1	2
CPU 1212C	DC/DC/DC	2	2	2
	AC/DC/RLS	0	1	2
	DC/DC/RLS	0	1	2
CPU 1214C	DC/DC/DC	2	2	2
	AC/DC/RLS	0	1	2
	DC/DC/RLS	0	1	2
CPU 1215C	DC/DC/DC	4	4	4
	AC/DC/RLS	0	1	2

Typ der CPU	Kein SB installiert	Mit einem SB (2 x DC-Ausgänge)	Mit einem SB (4 x DC-Ausgänge)
DC/DC/RLS	0	1	2

Tabelle 9- 23 Grenzfrequenzen von Impulsausgängen

Impulsausgang	Frequenz
Integriert	2 PTO: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$
	2 PTO: $2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
Standard-SB	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
Schnelle SBs (200 kHz)	MC-Anweisungen V2: $2 \text{ Hz} \leq f \leq 200 \text{ kHz}$
	MC-Anweisungen V1: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ <sup>1</sup>

<sup>1</sup> MC-Anweisungen V1 unterstützen eine maximale Frequenz von 100 kHz.

ACHTUNG
Die maximale Impulsfrequenz der Impulsgeneratoren beträgt 100 kHz für die digitalen Ausgänge der CPU, 20 kHz für die digitalen Ausgänge des Standard-Signalboards und 200 kHz für die digitalen Ausgänge der schnellen SBs (bzw. 100 kHz für MC-Anweisungen V1).

## Konfigurieren eines Impulsgenerators

1. Hinzufügen eines Technologieobjekts:
  - Erweitern Sie in der Projektnavigation den Knoten "Technologieobjekte" und wählen Sie "Neues Objekt hinzufügen" aus.
  - Wählen Sie das Symbol "Achse" (benennen Sie es ggf. um) und klicken Sie auf "OK", um den Konfigurationseditor für das Achsenobjekt zu öffnen.
  - Rufen Sie die Eigenschaften "PTO für Achssteuerung auswählen" unter "Grundparameter" auf und wählen Sie den gewünschten Impuls aus. Beachten Sie die zwei Ausgänge Q für Impuls und Richtung.

---

### Hinweis

Wenn PTO zuvor noch nicht in den CPU-Eigenschaften konfiguriert wurde, ist PTO so konfiguriert, dass einer der integrierten Ausgänge verwendet wird.

Wenn Sie ein Ausgangs-Signalboard verwenden, wählen Sie die Schaltfläche "Gerätekonfiguration", um die CPU-Eigenschaften aufzurufen. In den "Impulsoptionen" können Sie unter "Parametrierung" die Ausgangsquelle für einen Signalboard-Ausgang konfigurieren. "Impuls\_1" und "Impuls\_3" sind die einzigen verfügbaren Impulsausgänge auf dem Signalboard.

---

- Konfigurieren Sie die restlichen Grund- und erweiterten Parameter.
2. Programmieren Ihrer Anwendung: Fügen Sie die Anweisung MC\_Power in einen Codebaustein ein.
    - Für den Achseneingang wählen Sie das Technologieobjekt "Achse", das Sie angelegt und konfiguriert haben.
    - Wird der Eingang Enable auf WAHR gesetzt, können die anderen Bewegungssteuerungsanweisungen funktionieren.
    - Wird der Eingang Enable auf FALSCH gesetzt, werden die anderen Bewegungssteuerungsanweisungen storniert.

---

### Hinweis

Fügen Sie nur eine Anweisung MC\_Power pro Achse ein.

---

3. Fügen Sie die anderen Bewegungssteuerungsanweisungen ein, um die erforderliche Bewegung zu erzeugen.

---

### Hinweis

Konfigurieren eines Impulsgenerators für Signalboard-Ausgänge: Wählen Sie die Eigenschaften "Impulsgeneratoren (PTO/PWM)" für eine CPU (in der Gerätekonfiguration) und aktivieren Sie einen Impulsgenerator. Für jede S7-1200 CPU V1.0, V2.0, V2.1 und V2.2 sind zwei Impulsgeneratoren verfügbar, während bei der S7-1200 CPU V3.0 vier Impulsgeneratoren zur Verfügung stehen. In demselben Konfigurationsbereich wählen Sie unter "Impulsoptionen" für den Impulsgenerator die Verwendung "PTO" aus.

---

---

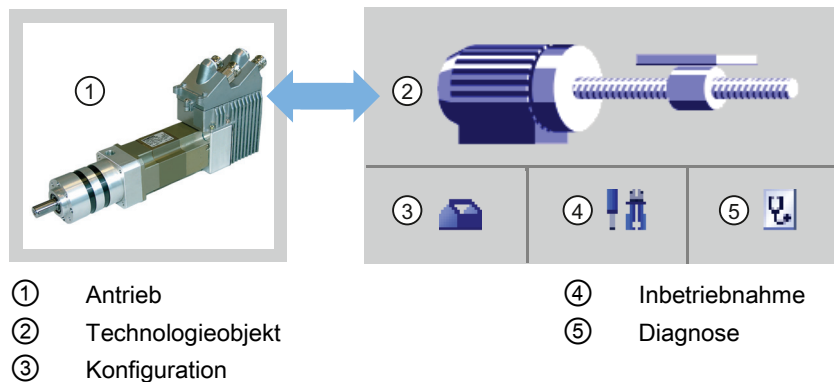
### Hinweis

Die CPU berechnet Bewegungssteuerungsaufgaben in "Scheiben" bzw. Segmenten von 10 ms. Wenn eine Scheibe ausgeführt wird, wartet die nächste Scheibe in der Warteschlange auf die Ausführung. Wenn Sie die Bewegungssteuerungsaufgabe einer Achse unterbrechen (indem Sie eine andere neue Bewegungssteuerungsaufgabe für die Achse ausführen), kann die neue Bewegungssteuerungsaufgabe maximal 20 ms lang nicht ausgeführt werden (die Restdauer der aktuellen Scheibe plus die Scheibe in der Warteschlange).

---

## 9.3.1 Achse konfigurieren

STEP 7 stellt die Konfigurationswerkzeuge, Inbetriebnahmewerkzeuge und Diagnosewerkzeuge für das Technologieobjekt "Achse" zur Verfügung.



---

### Hinweis

Die PTO benötigt die interne Funktionalität eines schnellen Zählers (HSC). Das bedeutet, dass der jeweilige schnelle Zähler nicht anderweitig verwendet werden kann.

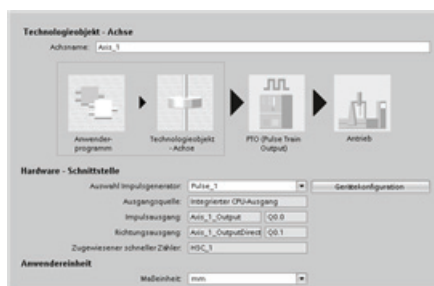
Die Zuordnung zwischen PTO und HSC ist fest vorgegeben. Wenn PTO1 aktiviert wird, erfolgt die Verbindung mit HSC1. Wenn PTO2 aktiviert wird, erfolgt die Verbindung mit HSC2. Dies gilt nur für S7-1200 V1.0-, V2.0-, V2.1- und V2.2-CPU's. S7-1200 V3.0-CPU's haben diese Einschränkung nicht.

Sie können den aktuellen Wert (z. B. in ID 1000) nicht überwachen, wenn Impulse auftreten.

---

Tabelle 9- 24 STEP 7-Werkzeuge für die Bewegungssteuerung

Werkzeug	Beschreibung
Konfiguration	<p>Konfiguriert die folgenden Eigenschaften des Technologieobjekts "Achse":</p> <ul style="list-style-type: none"> <li>• Auswahl des gewünschten PTO und Konfiguration der Antriebsschnittstelle</li> <li>• Eigenschaften der Mechanik und des Übertragungsverhältnisses des Antriebs (bzw. der Maschine oder Anlage)</li> <li>• Eigenschaften der Positionsgrenzwerte, Dynamik und Referenzpunktfahrt</li> </ul> <p>Speichern Sie die Konfiguration im Datenbaustein des Technologieobjekts.</p>
Inbetriebnahme	<p>Testet die Funktion Ihrer Achse, ohne dass ein Anwenderprogramm erstellt werden muss. Wenn das Werkzeug gestartet wird, wird das Steuerpanel angezeigt. Die folgenden Befehle sind im Steuerpanel verfügbar:</p> <ul style="list-style-type: none"> <li>• Achse freigeben und deaktivieren</li> <li>• Achse im Tipbetrieb bewegen</li> <li>• Achse absolut und relativ bewegen</li> <li>• Referenzpunktfahrt der Achse durchführen</li> <li>• Fehler quittieren</li> </ul> <p>Die Geschwindigkeit und die Beschleunigung/Verzögerung kann für die Bewegungssteuerungsbefehle angegeben werden. Das Steuerpanel zeigt auch den aktuellen Achsenzustand.</p>
Diagnose	<p>Überwachung des aktuellen Zustands und der Fehlerinformationen für die Achse und den Antrieb.</p>



Nachdem Sie das Technologieobjekt für die Achse angelegt haben, konfigurieren Sie die Achse, indem Sie die grundlegenden Parameter definieren, z. B. die PTO und die Konfiguration der Antriebsschnittstelle. Sie konfigurieren auch die anderen Eigenschaften der Achse wie Positionsgrenzwerte, Dynamik und Referenzpunktfahrt.

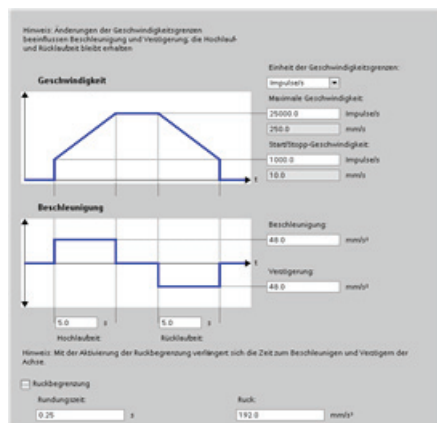
**ACHTUNG**

Sie müssen möglicherweise die Werte der Eingangsparameter der Bewegungssteuerungsanweisungen im Anwenderprogramm an die neue Einheit anpassen.

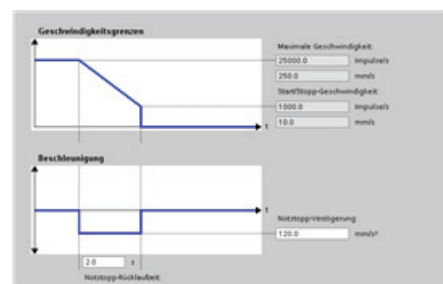




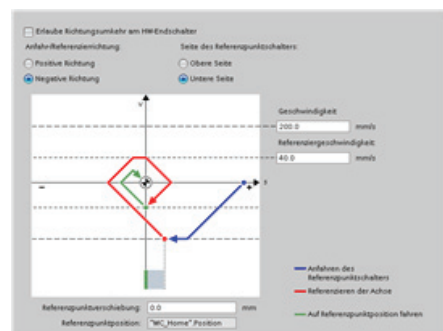
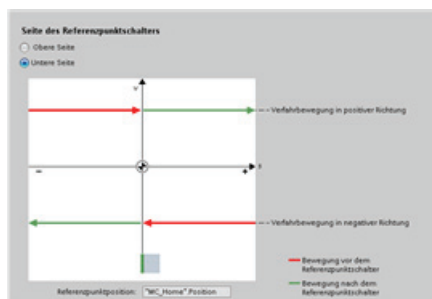
Konfigurieren Sie die Eigenschaften für die Antriebssignale, Antriebsmechanik und Positionsüberwachung (Hardware- und Software-Endschalter).




Sie konfigurieren die Bewegungssteuerungsdynamik und das Verhalten des Not-Aus-Befehls.



Ferner konfigurieren Sie das Verhalten der Referenzpunktfahrt (passiv und aktiv).



Im Steuerpanel "Inbetriebnahme" können Sie die Funktionalität unabhängig von Ihrem Anwenderprogramm testen.

 Klicken Sie auf das Symbol "Anlauf", um die Achse in Betrieb zu nehmen.

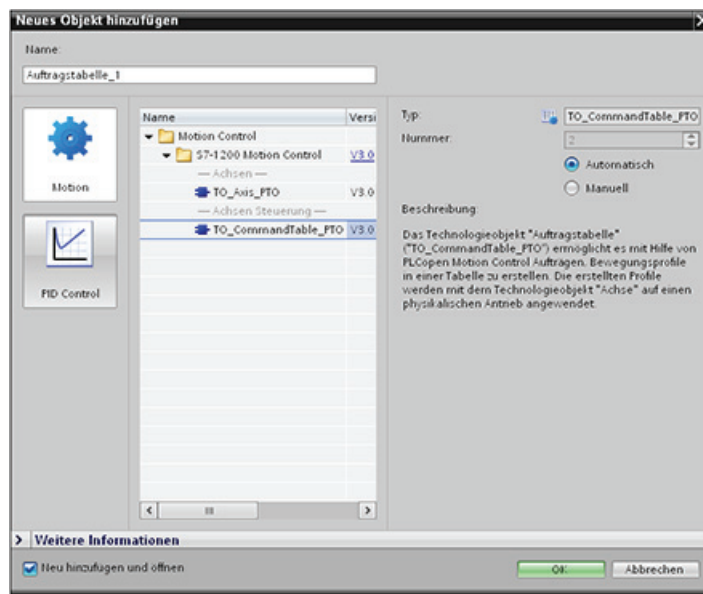
Das Steuerpanel zeigt den aktuellen Zustand der Achse. Sie können nicht nur die Achse aktivieren und deaktivieren, Sie können auch die Positionierung der Achse prüfen (sowohl absolut als auch relativ) und Sie können Geschwindigkeit, Beschleunigung und Verzögerung vorgeben. Weiterhin können Sie die Referenzpunktfahrt und den Tippbetrieb testen. Im Steuerpanel können Sie auch Fehler quittieren.

### 9.3.2 TO-Befehlstabelle PTO konfigurieren

Mit Hilfe der Technologieobjekte können Sie eine Anweisung CommandTable konfigurieren.

#### Hinzufügen eines Technologieobjekts

1. Erweitern Sie in der Projektnavigation den Knoten "Technologieobjekte" und wählen Sie "Neues Objekt hinzufügen" aus.
2. Wählen Sie das Symbol "Befehlstabelle" (benennen Sie es ggf. um) und klicken Sie auf "OK", um den Konfigurationseditor für das Befehlstabellenobjekt zu öffnen.



#### Schritte für Ihre Anwendung planen

Die gewünschte Bewegungsfolge können Sie im Konfigurationsfenster "Befehlstabelle" erstellen und das Ergebnis in der grafischen Darstellung im Kurvendiagramm prüfen.

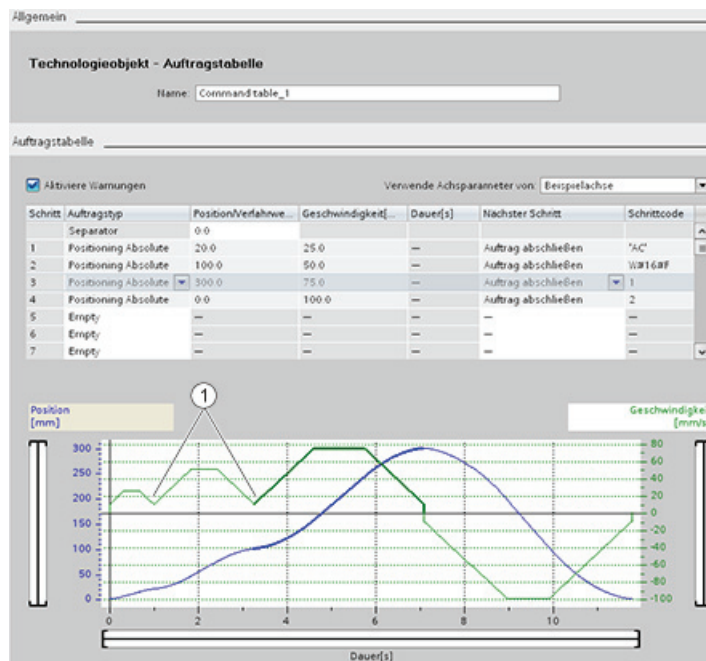
Sie können die Befehlstypen auswählen, die für die Verarbeitung der Befehlstabelle verwendet werden sollen. Bis zu 32 Schritte können eingegeben werden. Die Befehle werden der Reihe nach verarbeitet und erzeugen mühelos ein komplexes Bewegungsprofil.

Tabelle 9- 25 MC\_CommandTable-Befehlstypen

Befehlstyp	Beschreibung
Empty	Dieser Befehl dient als Platzhalter für jeden hinzuzufügenden Befehl. Der leere Eintrag wird bei der Verarbeitung der Befehlstabelle ignoriert.
Halt	Der Befehl hält die Achse an. Hinweis: Der Befehl wird nur nach einem Befehl "Velocity setpoint" durchgeführt.
Positioning Relative	Der Befehl positioniert die Achse basierend auf der Distanz. Er bewegt die Achse um die angegebene Distanz mit der angegebenen Geschwindigkeit.
Positioning Absolute	Der Befehl positioniert die Achse basierend auf der Lage. Er bewegt die Achse mit der vorgegebenen Geschwindigkeit an die angegebene Position.

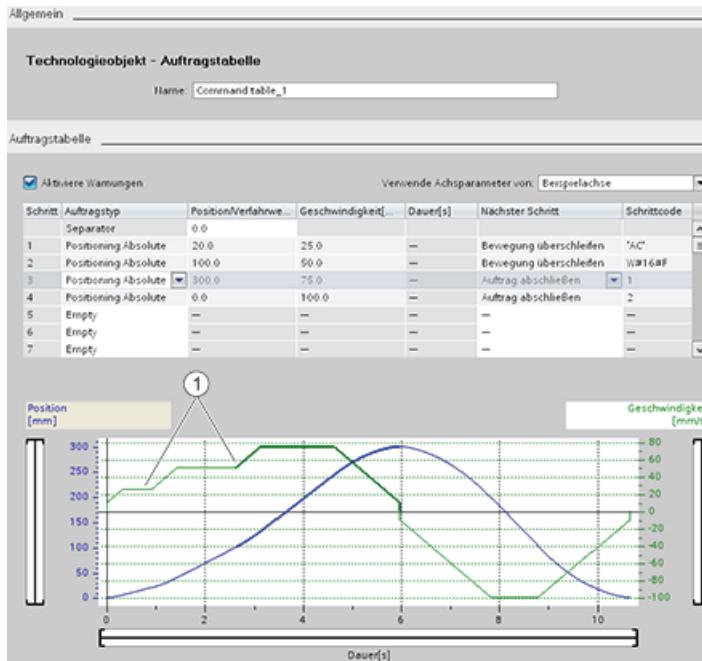
Befehlstyp	Beschreibung
Velocity setpoint	Der Befehl bewegt die Achse mit der angegebenen Geschwindigkeit.
Wait	Der Befehl wartet, bis der angegebene Zeitraum abgelaufen ist. "Wait" stoppt keine aktive Verfahrbewegung.
Separator	Der Befehl fügt eine Trennlinie ("Separator") oberhalb der ausgewählten Linie ein. Die Trennlinie ermöglicht die Definition mehrerer Profile in einer Befehlstabelle.

In der folgenden Abbildung wird "Command complete" als Übergang zum nächsten Schritt verwendet. Diese Art des Übergangs ermöglicht es Ihrem Gerät, auf die Start-/Stoppgeschwindigkeit zu verzögern und dann zu Beginn des nächsten Schritts erneut zu beschleunigen.



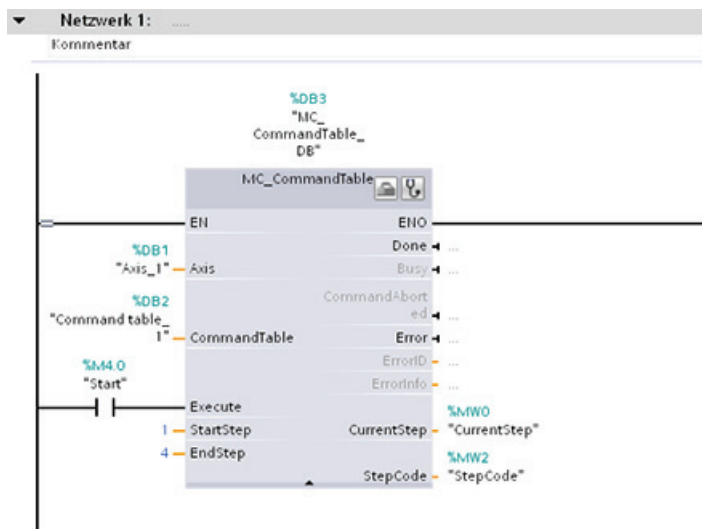
① Die Achse verzögert zwischen den Schritten auf die Start-/Stoppgeschwindigkeit.

In der folgenden Abbildung wird "Blending motion" als Übergang zum nächsten Schritt verwendet. Diese Art des Übergangs ermöglicht es Ihrem Gerät, die Geschwindigkeit bis zum Start des nächsten Schritts beizubehalten, was zu einem glatten Übergang des Geräts von einem Schritt zum nächsten führt. Bei Verwendung von "Blending motion" kann sich die Gesamtdauer für die vollständige Ausführung eines Profils verkürzen. Ohne Blending dauert die Ausführung des Beispiels sieben Sekunden. Mit Blending wird die Ausführungszeit um eine Sekunde auf insgesamt sechs Sekunden reduziert.



- ① Die Achse bewegt sich ununterbrochen weiter und beschleunigt oder verzögert auf die Geschwindigkeit des nächsten Schritts, wodurch Zeit gespart und mechanischer Verschleiß verringert wird.

Die Funktionsweise Ihrer Befehlstabelle wird, wie im Folgenden gezeigt, von der Anweisung MC\_CommandTable gesteuert:



### 9.3.3 Bewegungssteuerungsanweisungen

#### Hinweis

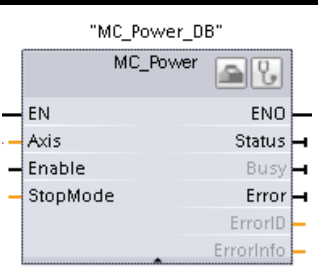
Die CPU berechnet Bewegungssteuerungsaufgaben in "Scheiben" bzw. Segmenten von 10 ms. Wenn eine Scheibe ausgeführt wird, wartet die nächste Scheibe in der Warteschlange auf die Ausführung. Wenn Sie die Bewegungssteuerungsaufgabe einer Achse unterbrechen (indem Sie eine andere neue Bewegungssteuerungsaufgabe für die Achse ausführen), kann die neue Bewegungssteuerungsaufgabe maximal 20 ms lang nicht ausgeführt werden (die Restdauer der aktuellen Scheibe plus die Scheibe in der Warteschlange).

#### 9.3.3.1 Anweisung MC\_Power

#### ACHTUNG

Wenn die Achse wegen eines Fehlers ausgeschaltet wird, wird sie nach Behebung und Quittierung des Fehlers automatisch wieder aktiviert. Hierfür ist erforderlich, dass der Eingangsparameter Enable den Wert WAHR während dieses Vorgangs gespeichert hat.

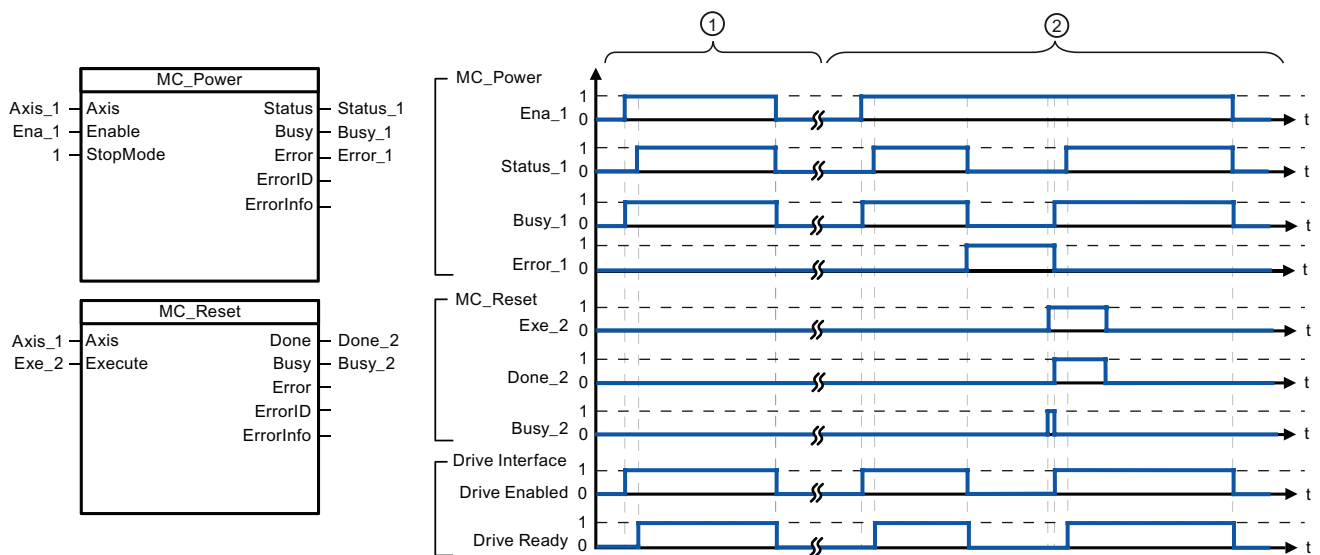
Tabelle 9- 26 Anweisung MC\_Power

KOP/FUP	SCL	Beschreibung
 <p>The screenshot shows a graphical representation of the MC_Power DB. It has several input and output ports. Inputs include EN, Axis, Enable, and StopMode. Outputs include END, Status, Busy, Error, ErrorID, and ErrorInfo. There are also some status icons at the top right of the DB box.</p>	<pre>"MC_Power_DB" (   Axis:=_multi_fb_in_,   Enable:=_bool_in_,   StopMode:=_int_in_,   Status=&gt;_bool_out_,   Busy=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Die Bewegungssteuerungsanweisung MC_Power aktiviert oder deaktiviert eine Achse. Bevor Sie die Achse aktivieren oder deaktivieren können, prüfen Sie die folgenden Bedingungen:</p> <ul style="list-style-type: none"> <li>• Das Technologieobjekt wurde korrekt konfiguriert.</li> <li>• Es steht kein Freigabe verhindernder Fehler an.</li> </ul> <p>Die Ausführung von MC_Power kann nicht von einer Bewegungssteuerungsaufgabe abgebrochen werden. Durch Deaktivieren der Achse (Eingangsparameter Enable = FALSCH) werden alle Bewegungssteuerungsaufgaben für das zugehörige Technologieobjekt abgebrochen.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_Power\_DB" der Name des Instanz-DBs.

Tabelle 9- 27 Parameter für die MC\_Power-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Enable	IN	Bool	<ul style="list-style-type: none"> <li>FALSE (Standard): Alle aktiven Aufgaben werden entsprechend dem parametrisierten "StopMode" abgebrochen und die Achse wird gestoppt.</li> <li>TRUE: Die Bewegungssteuerung versucht, die Achse zu aktivieren.</li> </ul>
StopMode	IN	Int	<ul style="list-style-type: none"> <li>0: Not-Aus: Steht eine Anforderung zur Deaktivierung der Achse an, bremst die Achse mit der konfigurierten Notfallverzögerung. Die Achse wird nach Erreichen des Stillstands deaktiviert.</li> <li>1: Sofortiger Stopp: Steht eine Anforderung zur Deaktivierung der Achse an, wird die Achse ohne Verzögerung deaktiviert. Der Impulsausgang wird sofort gestoppt.</li> <li>2: Not-Aus mit Rucksteuerung: Steht eine Anforderung zur Deaktivierung der Achse an, bremst die Achse mit der konfigurierten Not-Aus-Verzögerung. Wenn die Rucksteuerung aktiviert ist, wird der konfigurierte Ruck berücksichtigt. Die Achse wird nach Erreichen des Stillstands deaktiviert.</li> </ul>
Status	OUT	Bool	<p>Status der Achsenfreigabe:</p> <ul style="list-style-type: none"> <li>FALSE: Die Achse ist deaktiviert: <ul style="list-style-type: none"> <li>Die Achse führt keine Bewegungssteuerungsaufgaben aus und nimmt keine neuen Aufgaben an (Ausnahme: Aufgabe MC_Reset).</li> <li>Die Achse ist nicht am Referenzpunkt.</li> <li>Beim Deaktivieren wechselt der Zustand erst dann nach FALSCH, wenn die Achse den Stillstand erreicht.</li> </ul> </li> <li>TRUE: Die Achse ist aktiviert: <ul style="list-style-type: none"> <li>Die Achse ist bereit, Bewegungssteuerungsaufgaben auszuführen.</li> <li>Beim Aktivieren der Achse wechselt der Zustand erst dann nach WAHR, wenn das Signal "Antrieb bereit" ansteht. Wurde die Antriebsschnittstelle "Antrieb bereit" während der Achsenkonfiguration nicht eingerichtet, wechselt der Zustand sofort nach WAHR.</li> </ul> </li> </ul>
Busy	OUT	Bool	<p>FALSE: MC_Power ist nicht aktiv. TRUE: MC_Power ist aktiv.</p>
Error	OUT	Bool	<p>FALSE: Kein Fehler TRUE: In der Bewegungssteuerungsanweisung "MC_Power" oder im zugehörigen Technologieobjekt ist ein Fehler aufgetreten. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.</p>
ErrorID	OUT	Word	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter ErrorID



- ① Eine Achse wird aktiviert und dann wieder deaktiviert. Nachdem der Antrieb das Signal "Antrieb bereit" an die CPU zurückgemeldet hat, kann die erfolgreiche Aktivierung über "Status\_1" ausgelesen werden.
- ② Nach einer Achsenfreigabe ist ein Fehler aufgetreten, der verursacht hat, dass die Achse deaktiviert wurde. Der Fehler wird behoben und mit "MC\_Reset" quittiert. Die Achse wird dann wieder aktiviert.

Um eine Achse mit konfigurierter Antriebsschnittstelle zu aktivieren, gehen Sie wie folgt vor:

1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Initialisieren Sie den Eingangsparameter "StopMode" mit dem gewünschten Wert. Setzen Sie den Eingangsparameter "Enable" auf WAHR.

Der Freigabeausgang für "Antrieb freigegeben" wechselt nach WAHR, um die Spannung für den Antrieb zu aktivieren. Die CPU wartet auf das Signal "Antrieb bereit" des Antriebs.

Wenn das Signal "Antrieb bereit" am konfigurierten Bereitschaftseingang der CPU verfügbar ist, wird die Achse freigegeben. Der Ausgangsparameter "Status" und die Technologieobjektvariable <Achsenname>.StatusBits.Enable geben den Wert WAHR an.

Um eine Achse ohne konfigurierte Antriebsschnittstelle zu aktivieren, gehen Sie wie folgt vor:


1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Initialisieren Sie den Eingangsparameter "StopMode" mit dem gewünschten Wert. Setzen Sie den Eingangsparameter "Enable" auf WAHR. Die Achse ist aktiviert. Der Ausgangsparameter "Status" und die Technologieobjektvariable <Achsenname>.StatusBits.Enable geben den Wert WAHR an.

Um eine Achse zu deaktivieren, gehen Sie wie folgt vor:

1. Bringen Sie die Achse zum Stillstand.  
Anhand der Technologieobjektvariable <Achsenname>.StatusBits.StandStill können Sie erkennen, wann sich die Achse im Stillstand befindet.
2. Setzen Sie den Eingangsparameter "Enable" nach Erreichen des Stillstands auf FALSCH.
3. Wenn die Ausgangsparameter "Busy" und "Status" und die Technologieobjektvariable <Achsenname>.StatusBits.Enable den Wert FALSCH anzeigen, ist die Deaktivierung der Achse beendet.

### 9.3.3.2 Anweisung MC\_Reset

Tabelle 9- 28 Anweisung MC\_Reset

KOP/FUP	SCL	Beschreibung
	<pre>"MC_Reset_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Restart:=_bool_in_,   Done=&gt;_bool_out_,   Busy=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Mit der Anweisung MC_Reset quittieren Sie "Betriebsfehler mit Achsenstopp" und "Konfigurationsfehler". Die Fehler, die quittiert werden müssen, finden Sie in der "Liste von ErrorIDs und ErrorInfos" unter "Abhilfe".</p> <p>Bevor Sie die Anweisung MC_Reset verwenden, müssen Sie die Ursache eines anstehenden zu quittierenden Konfigurationsfehlers behoben haben (indem Sie beispielsweise einen ungültige Beschleunigungswert im Technologieobjekt "Achse" in einen gültigen Wert ändern).</p> <p>Ab Version 3.0 ermöglicht der Befehl Restart das Laden der Achsenkonfiguration in den Arbeitsspeicher im Betriebszustand RUN.</p>

- STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "MC\_Reset\_DB" der Name des Instanz-DBs.

Die Aufgabe MC\_Reset kann von keiner anderen Bewegungssteuerungsaufgabe abgebrochen werden. Die neue Aufgabe MC\_Reset bricht keine anderen aktiven Bewegungssteuerungsaufgaben ab.

Tabelle 9- 29 Parameter der Anweisung MC\_Reset

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke
Restart	IN	Bool	TRUE = Laden der Achsenkonfiguration aus dem Ladespeicher in den Arbeitsspeicher. Der Befehl kann nur bei deaktivierter Achse ausgeführt werden.
			FALSE = Anstehende Fehler werden quittiert
Done	OUT	Bool	WAHR = Fehler wurde quittiert.
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUTP	Word	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter ErrorID

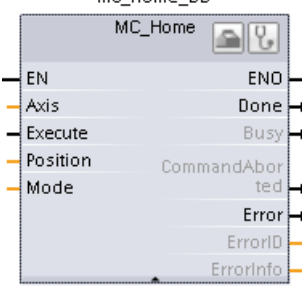


Um einen Fehler mit MC\_Reset zu quittieren, gehen Sie folgendermaßen vor:

1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Starten Sie die Quittierung des Fehlers mit einer steigenden Flanke am Eingangsparameter Execute.
3. Der Fehler wurde quittiert, wenn Done gleich WAHR ist und die Technologieobjektvariable <Achsenname>.StatusBits.Error gleich FALSCH ist.

### 9.3.3.3 Anweisung MC\_Home

Tabelle 9- 30 Anweisung MC\_Home

KOP/FUP	SCL	Beschreibung
	<pre>"MC_Home_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Position:=_real_in_,   Mode:=_int_in_,   Done=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Mit der Anweisung MC_Home passen Sie die Achsenkoordinaten an die reale, physikalische Position des Antriebs an. Für die absolute Positionierung der Achse ist eine Referenzpunktfahrt erforderlich:</p> <p>Um die Anweisung MC_Home zu verwenden, muss die Achse zunächst freigegeben werden.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_Home\_DB" der Name des Instanz-DBs.

Die folgenden Arten von Referenzpunktfahrten stehen zur Verfügung:

- Direkte Referenzpunktfahrt absolut (Mode = 0): Die aktuelle Achsenposition wird auf den Wert des Parameters "Position" gesetzt.
- Direkte Referenzpunktfahrt relativ (Mode = 1): Die aktuelle Achsenposition wird um den Wert des Parameters "Position" versetzt.
- Passive Referenzpunktfahrt (Mode = 2): Während der passiven Referenzpunktfahrt führt die Anweisung MC\_Home keine Referenzpunktfahrtbewegung durch. Die für diesen Schritt erforderliche Verfahrbewegung müssen Sie über andere Bewegungssteuerungsanweisungen implementieren. Wenn der Referenzpunktwechsel erkannt wird, wird die Achse an den Referenzpunkt gefahren.
- Aktive Referenzpunktfahrt (Mode = 3): Die Referenzpunktfahrt wird automatisch durchgeführt.

Tabelle 9- 31 Parameter für die MC\_Home-Anweisung

Parameter und Datentyp		Datentyp	Beschreibung
Axis	IN	TO_Axis_PTO	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke
Position	IN	Real	<ul style="list-style-type: none"> <li>Mode = 0, 2 und 3 (Absolute Position der Achse nach Beendigung der Referenzpunktfahrt)</li> <li>Mode = 1 (Korrekturwert für die aktuelle Achsenposition)</li> </ul> Grenzwerte: $-1,0e^{12} \leq \text{Position} \leq 1,0e^{12}$
Mode	IN	Int	Art der Referenzpunktfahrt <ul style="list-style-type: none"> <li>0: Direkte Referenzpunktfahrt absolut Die neue Achsenposition ist der Positionswert des Parameters "Position".</li> <li>1: Direkte Referenzpunktfahrt relativ Die neue Achsenposition ist die aktuelle Achsenposition + Positionswert des Parameters "Position".</li> <li>2: Passive Referenzpunktfahrt Referenzpunktfahrt entsprechend der Achsenkonfiguration. Nach der Referenzpunktfahrt wird der Wert des Parameters "Position" als neue Achsenposition eingestellt.</li> <li>3: Aktive Referenzpunktfahrt Referenzpunktannäherung entsprechend der Achsenkonfiguration. Nach der Referenzpunktfahrt wird der Wert des Parameters "Position" als neue Achsenposition eingestellt.</li> </ul>
Done	OUT	Bool	WAHR = Aufgabe durchgeführt
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter ErrorID

**Hinweis**

**Die Referenzpunktfahrt der Achse geht unter den folgenden Voraussetzungen verloren**

- Deaktivierung der Achse durch die Anweisung MC\_Power
- Umschalten zwischen Automatik- und Handbetrieb
- Nach dem Start der aktiven Referenzpunktfahrt (nach erfolgreicher Durchführung der Referenzpunktfahrt steht die Referenzpunktfahrt der Achse erneut zur Verfügung.)
- Nach Aus- und wieder Einschalten der CPU
- Nach Neustart der CPU (RUN-in-STOP oder STOP-in-RUN)

Um die Achse an den Referenzpunkt zu fahren, gehen Sie folgendermaßen vor:

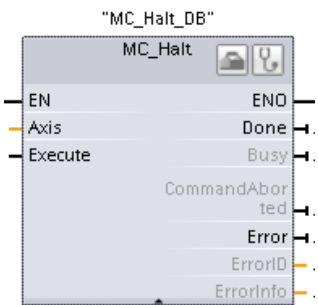
1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Initialisieren Sie die erforderlichen Eingangsparameter mit Werten und starten Sie die Referenzpunktfahrt mit einer steigenden Flanke am Eingangsparameter "Execute".
3. Wenn der Ausgangsparameter "Done" und die Technologieobjektvariable <Achsenname>.StatusBits.HomingDone den Wert WAHR angeben, ist die Referenzpunktfahrt beendet.

Tabelle 9- 32 Übersteuerungsantwort

Betriebsart	Beschreibung	
0 oder 1	Die Aufgabe MC_Home kann von keiner anderen Bewegungssteuerungsaufgabe abgebrochen werden. Die neue Aufgabe MC_Home bricht keine aktiven Bewegungssteuerungsaufgaben ab. Positionsbezogene Bewegungssteuerungsaufgaben werden nach der Referenzpunktfahrt entsprechend der neuen Referenzpunktposition (Wert am Eingangsparameter Position) wieder aufgenommen.	
2	Die Aufgabe MC_Home kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden: Mode der Aufgabe MC_Home = 2, 3: Die neue Aufgabe MC_Home bricht die folgende aktive Bewegungssteuerungsaufgabe ab. Mode der Aufgabe MC_Home = 2: Positionsbezogene Bewegungssteuerungsaufgaben werden nach der Referenzpunktfahrt entsprechend der neuen Referenzpunktposition (Wert am Eingangsparameter Position) wieder aufgenommen.	
3	Die Aufgabe MC_Home kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden: <ul style="list-style-type: none"> <li>• MC_Home Mode = 3</li> <li>• MC_Halt</li> <li>• MC_MoveAbsolute</li> <li>• MC_MoveRelative</li> <li>• MC_MoveVelocity</li> <li>• MC_MoveJog</li> </ul>	Die neue Aufgabe MC_Home bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab: <ul style="list-style-type: none"> <li>• MC_Home-Modus = 2, 3</li> <li>• MC_Halt</li> <li>• MC_MoveAbsolute</li> <li>• MC_MoveRelative</li> <li>• MC_MoveVelocity</li> <li>• MC_MoveJog</li> </ul>

### 9.3.3.4 Anweisung MC\_Halt

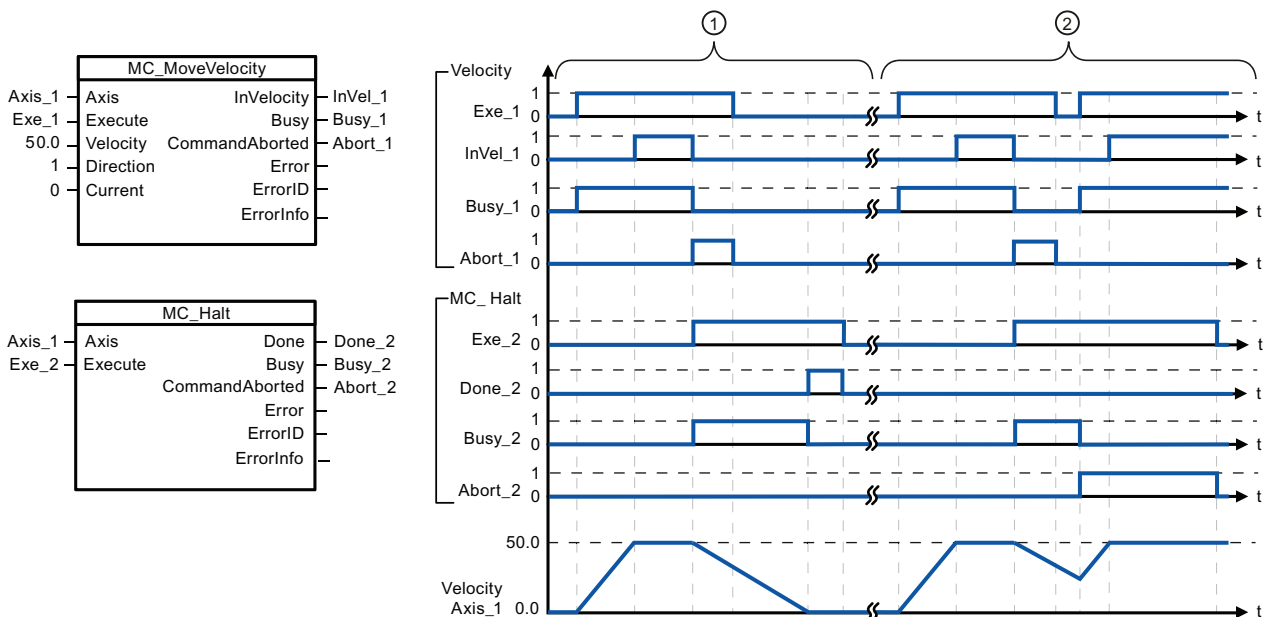
Tabelle 9- 33 Anweisung MC\_Halt

KOP/FUP	SCL	Beschreibung
	<pre>"MC_Halt_DB" (   Axis:= _multi_fb_in_,   Execute:= _bool_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_);</pre>	<p>Mit der Anweisung MC_Halt stoppen Sie jegliche Bewegung und bringen die Achse zum Stillstand. Die Stillstandposition ist nicht definiert.</p> <p>Um die Anweisung MC_Halt zu verwenden, muss die Achse zunächst freigegeben werden.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_Halt\_DB" der Name des Instanz-DBs.

Tabelle 9- 34 Parameter für die MC\_Halt-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke
Done	OUT	Bool	WAHR = Nullgeschwindigkeit erreicht
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter ErrorID



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 5,0

- ① Die Achse wird von einer Aufgabe MC\_Halt gebremst, bis sie zum Stillstand kommt. Der Stillstand der Achse wird über "Done\_2" gemeldet.
- ② Während die Aufgabe MC\_Halt die Achse bremst, wird die Aufgabe von einer anderen Bewegungssteuerungsaufgabe abgebrochen. Der Abbruch wird über "Abort\_2" gemeldet.

### Übersteuerungsantwort

Die Aufgabe MC\_Halt kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

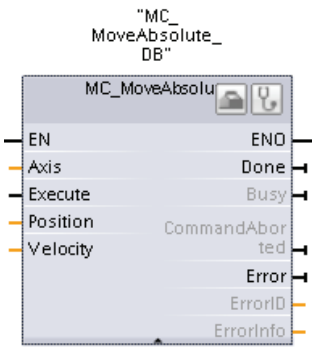
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

Die neue Aufgabe MC\_Halt bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

9.3.3.5 Anweisung MC\_MoveAbsolute

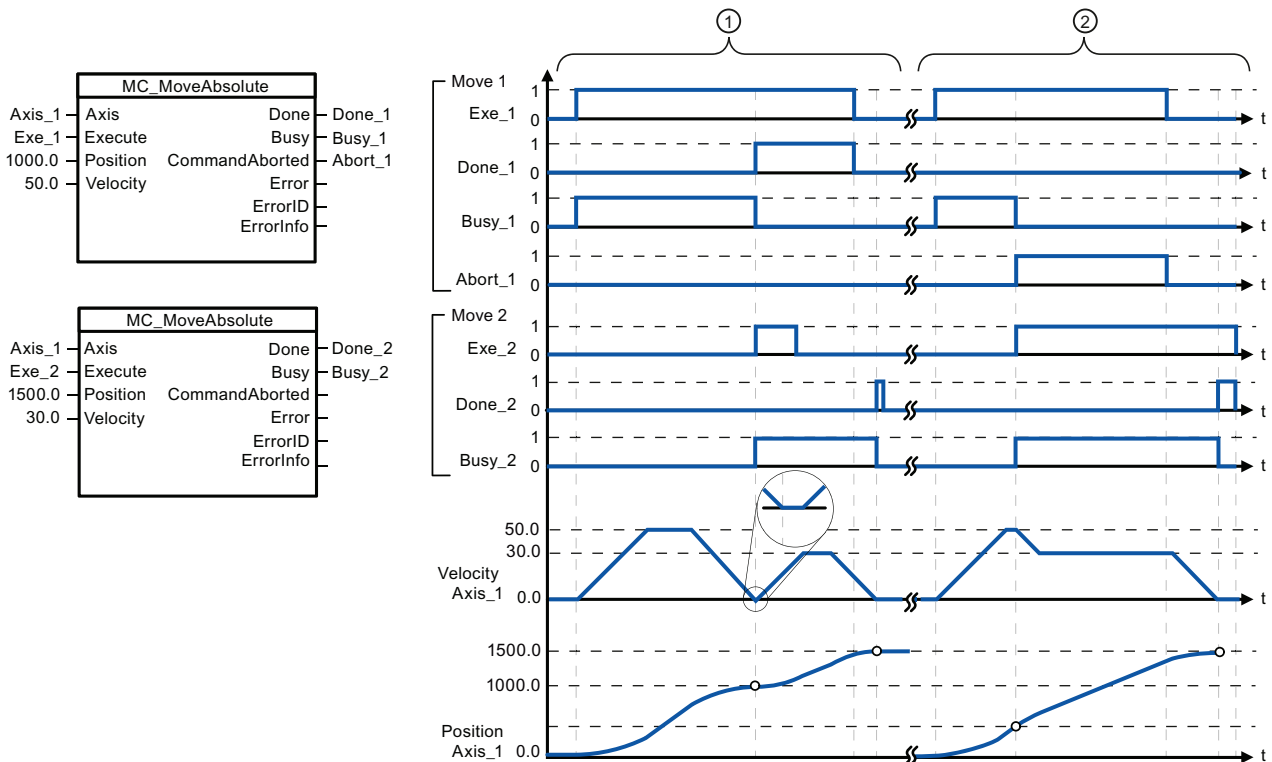
Tabelle 9- 35 Anweisung MC\_MoveAbsolute

KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveAbsolute_DB" (   Axis:= _multi_fb_in_,   Execute:= _bool_in_,   Position:= _real_in_,   Velocity:= _real_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_);</pre>	<p>Mit der Anweisung MC_MoveAbsolute starten Sie eine Positionierbewegung der Achse zu einer absoluten Position.</p> <p>Um die Anweisung MC_MoveAbsolute zu verwenden, muss die Achse zunächst freigegeben und zum Referenzpunkt gefahren werden.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_MoveAbsolute\_DB" der Name des Instanz-DBs.

Tabelle 9- 36 Parameter für die MC\_MoveAbsolute-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke (Standardwert: Falsch)
Position	IN	Real	Absolute Zielposition (Standardwert: 0.0) Grenzwerte: $-1,0e^{12} \leq \text{Position} \leq 1,0e^{12}$
Velocity	IN	Real	Achsen­geschwindigkeit (Standardwert: 10.0) Diese Geschwindigkeit wird wegen der konfigurierten Beschleunigung und Verzögerung und der anzufahrenden Zielposition nicht immer erreicht. Grenzwerte: Start-/Stoppgeschwindigkeit $\leq$ Velocity $\leq$ Maximalgeschwindigkeit
Done	OUT	Bool	WAHR = Absolute Zielposition erreicht
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error" (Standardwert: 0000)
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter "ErrorID" (Standardwert: 0000)



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 10,0

- ① Eine Achse wird mit der Aufgabe MC\_MoveAbsolute zur absoluten Position 1000,0 gefahren. Wenn die Achse die Zielposition erreicht, wird dies über "Done\_1" gemeldet. Wenn "Done\_1" = WAHR ist, wird eine andere Aufgabe MC\_MoveAbsolute mit der Zielposition 1500,0 gestartet. Wegen der Antwortzeiten (z. B. Zykluszeit des Anwenderprogramms usw.) kommt die Achse kurz zum Stillstand (siehe vergrößerter Ausschnitt). Wenn die Achse die neue Zielposition erreicht, wird dies über "Done\_2" gemeldet.
- ② Eine aktive Aufgabe MC\_MoveAbsolute wird von einer anderen Aufgabe MC\_MoveAbsolute abgebrochen. Der Abbruch wird über "Abort\_1" gemeldet. Die Achse wird dann mit der neuen Geschwindigkeit zur neuen Zielposition 1500,0 gefahren. Wenn die neue Zielposition erreicht ist, wird dies über "Done\_2" gemeldet.

### Übersteuerungsantwort

Die Aufgabe MC\_MoveAbsolute kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

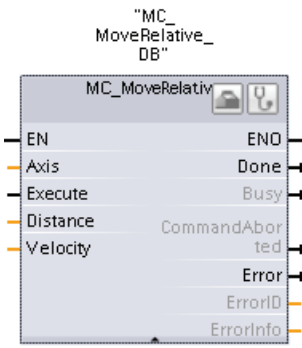
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

Die neue Aufgabe MC\_MoveAbsolute bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

### 9.3.3.6 Anweisung MC\_MoveRelative

Tabelle 9- 37 Anweisung MC\_MoveRelative

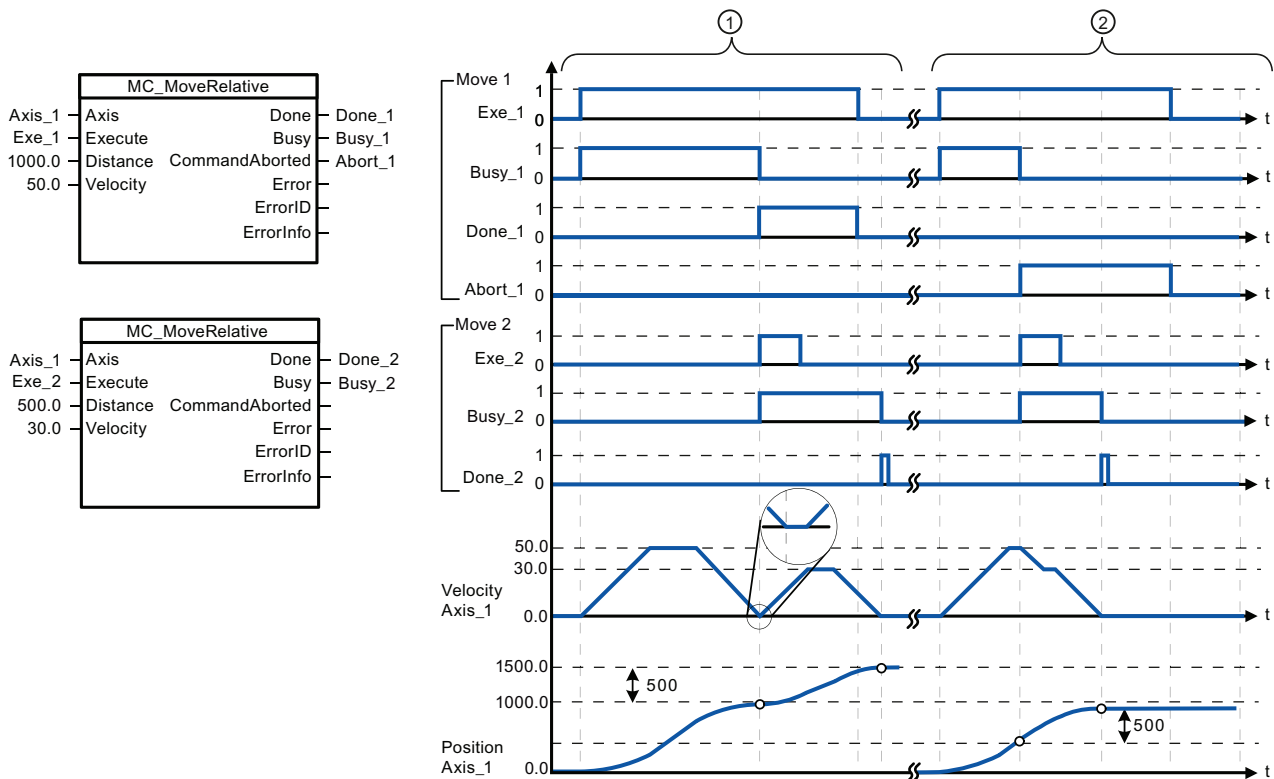
KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveRelative_DB" (   Axis:= _multi_fb_in_,   Execute:= _bool_in_,   Distance:= _real_in_,   Velocity:= _real_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_);</pre>	<p>Mit der Anweisung MC_MoveRelative starten Sie eine Positionierbewegung relativ zur Startposition.</p> <p>Um die Anweisung MC_MoveRelative zu verwenden, muss die Achse zunächst freigegeben werden.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_MoveRelative\_DB" der Name des Instanz-DBs.

Tabelle 9- 38 Parameter für die MC\_MoveRelative-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke (Standardwert: Falsch)
Distance	IN	Real	Verfahrweg für den Positioniervorgang (Standardwert: 0.0) Grenzwerte: $-1,0e^{12} \leq \text{Distance} \leq 1,0e^{12}$
Velocity	IN	Real	Achsgeschwindigkeit (Standardwert: 10.0) Diese Geschwindigkeit wird wegen der konfigurierten Beschleunigung und Verzögerung und des zu fahrenden Wegs nicht immer erreicht. Grenzwerte: Start-/Stoppgeschwindigkeit $\leq \text{Velocity} \leq$ Maximalgeschwindigkeit
Done	OUT	Bool	WAHR = Zielposition erreicht
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error" (Standardwert: 0000)
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter "ErrorID" (Standardwert: 0000)





Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 10,0

- ① Die Achse wird von einer Aufgabe MC\_MoveRelative den Weg ("Distance") 1000,0 gefahren. Wenn die Achse die Zielposition erreicht, wird dies über "Done\_1" gemeldet. Wenn "Done\_1" = WAHR ist, wird eine andere Aufgabe MC\_MoveRelative mit dem Verfahrensweg 500,0 gestartet. Wegen der Antwortzeiten (z. B. Zykluszeit des Anwenderprogramms) kommt die Achse kurz zum Stillstand (siehe vergrößerter Ausschnitt). Wenn die Achse die neue Zielposition erreicht, wird dies über "Done\_2" gemeldet.
- ② Eine aktive Aufgabe MC\_MoveRelative wird von einer anderen Aufgabe MC\_MoveRelative abgebrochen. Der Abbruch wird über "Abort\_1" gemeldet. Die Achse wird dann mit der neuen Geschwindigkeit den neuen Weg ("Distance") 500,0 gefahren. Wenn die neue Zielposition erreicht ist, wird dies über "Done\_2" gemeldet.

### Übersteuerungsantwort

Die Aufgabe MC\_MoveRelative kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

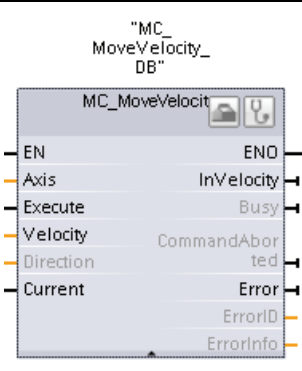
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

Die neue Aufgabe MC\_MoveRelative bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

### 9.3.3.7 Anweisung MC\_MoveVelocity

Tabelle 9- 39 Anweisung MC\_MoveVelocity

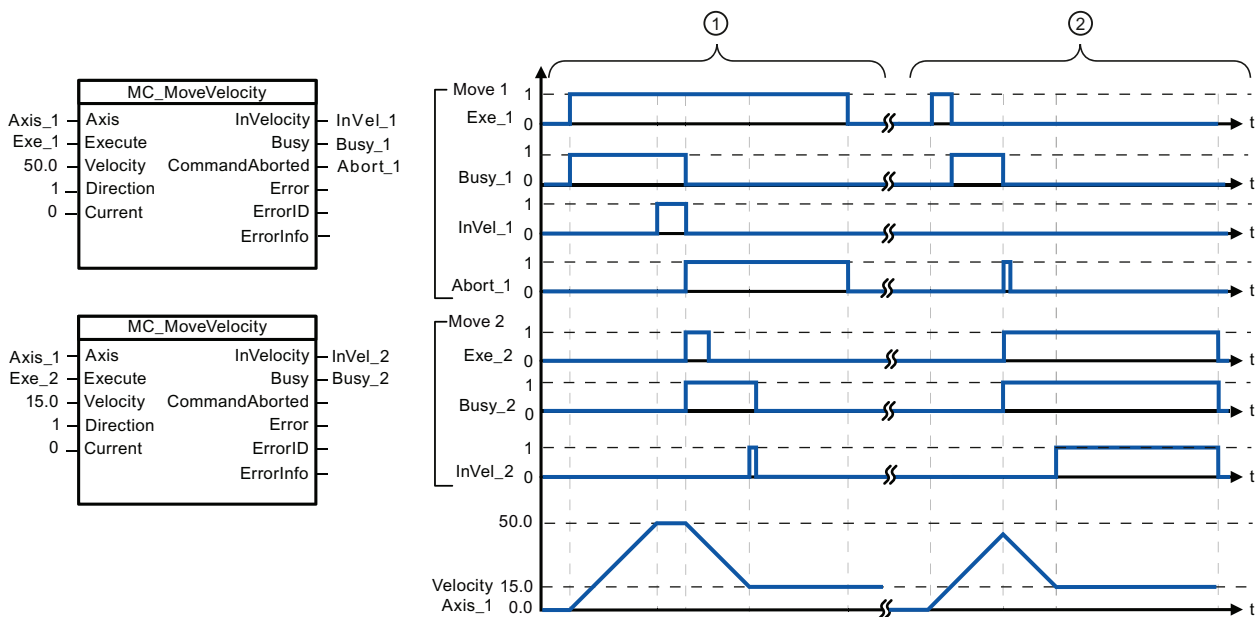
KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveVelocity_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Velocity:=_real_in_,   Direction:=_int_in_,   Current:=_bool_in_,   InVelocity=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Mit der Anweisung MC_MoveVelocity bewegen Sie die Achse konstant mit der angegebenen Geschwindigkeit.</p> <p>Um die Anweisung MC_MoveVelocity zu verwenden, muss die Achse zunächst freigegeben werden.</p>

- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "MC\_MoveVelocity\_DB" der Name des Instanz-DBs.

Tabelle 9- 40 Parameter für die MC\_MoveVelocity-Anweisung

Parameter und Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1 Technologieobjekt "Achse"
Execute	IN	Bool Starten der Aufgabe bei einer positiven Flanke (Standardwert: Falsch)
Velocity	IN	Real Geschwindigkeitsangabe für die Achsenbewegung (Standardwert: 10.0) Grenzwerte: Start-/Stoppgeschwindigkeit ≤  Velocity  ≤ Maximalgeschwindigkeit (Velocity = 0,0 ist zulässig)
Direction	IN	Int Richtungsangabe: <ul style="list-style-type: none"> <li>0: Die Drehrichtung entspricht dem Vorzeichen des Werts im Parameter "Velocity" (Standardwert)</li> <li>1: Positive Drehrichtung (das Vorzeichen des Werts im Parameter "Velocity" wird ignoriert).</li> <li>2: Negative Drehrichtung (das Vorzeichen des Werts im Parameter "Velocity" wird ignoriert).</li> </ul>

Parameter und Datentyp		Datentyp	Beschreibung
Current	IN	Bool	<p>Aktuelle Geschwindigkeit beibehalten:</p> <ul style="list-style-type: none"> <li>• FALSCH: "Aktuelle Geschwindigkeit beibehalten" ist deaktiviert. Die Werte der Parameter "Velocity" und "Direction" werden verwendet. (Standardwert)</li> <li>• WAHR: "Aktuelle Geschwindigkeit beibehalten" ist aktiviert. Die Werte der Parameter "Velocity" und "Direction" werden nicht berücksichtigt.</li> </ul> <p>Wenn die Achse die Bewegung mit der aktuellen Geschwindigkeit wieder aufnimmt, gibt der Parameter "InVelocity" den Wert TRUE aus.</p>
InVelocity	OUT	Bool	<p>WAHR:</p> <ul style="list-style-type: none"> <li>• Wenn "Current" = FALSE: Die im Parameter "Velocity" angegebene Geschwindigkeit wurde erreicht.</li> <li>• Wenn "Current" = TRUE: Die Achse fährt zur Startzeit mit der aktuellen Geschwindigkeit.</li> </ul>
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error" (Standardwert: 0000)
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter "ErrorID" (Standardwert: 0000)



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 10,0

- ① Eine aktive Aufgabe MC\_MoveVelocity meldet über "InVel\_1", dass die Zielgeschwindigkeit erreicht wurde. Sie wird dann von einer anderen Aufgabe MC\_MoveVelocity abgebrochen. Der Abbruch wird über "Abort\_1" gemeldet. Wenn die neue Zielgeschwindigkeit 15,0 erreicht ist, wird dies über "InVel\_2" gemeldet. Die Achse bewegt sich dann mit der neuen konstanten Geschwindigkeit weiter.
- ② Eine aktive Aufgabe MC\_MoveVelocity wird von einer anderen Aufgabe MC\_MoveVelocity abgebrochen, bevor sie ihre Zielgeschwindigkeit erreicht. Der Abbruch wird über "Abort\_1" gemeldet. Wenn die neue Zielgeschwindigkeit 15,0 erreicht ist, wird dies über "InVel\_2" gemeldet. Die Achse bewegt sich dann mit der neuen konstanten Geschwindigkeit weiter.

### Übersteuerungsantwort

Die Aufgabe MC\_MoveVelocity kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

Die neue Aufgabe MC\_MoveVelocity bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

**Hinweis**

**Verhalten bei auf Null gesetzter Geschwindigkeit (Velocity = 0,0)**

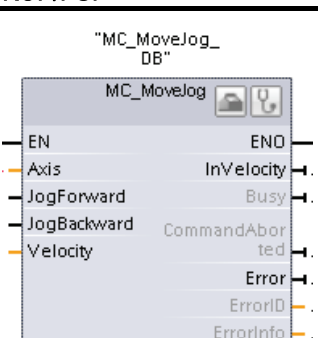
Eine Aufgabe MC\_MoveVelocity mit "Velocity" = 0,0 (wie eine Aufgabe MC\_Halt) bricht aktive Bewegungssteuerungsaufgaben ab und stoppt die Achse mit der konfigurierten Verzögerung. Wenn die Achse zum Stillstand kommt, gibt der Ausgangsparameter "InVelocity" mindestens einen Programmzyklus lang WAHR an.

"Busy" gibt während des Verzögerungsvorgangs den Wert WAHR an und geht zusammen mit "InVelocity" nach FALSCH. Wenn der Parameter "Execute" = WAHR gesetzt ist, sind "InVelocity" und "Busy" als gespeichert gesetzt.

Wenn die Aufgabe MC\_MoveVelocity gestartet wird, wird das Statusbit "SpeedCommand" im Technologieobjekt gesetzt. Das Statusbit "ConstantVelocity" wird beim Stillstand der Achse gesetzt. Beide Bits werden an die neue Situation angepasst, wenn eine neue Bewegungssteuerungsaufgabe gestartet wird.

**9.3.3.8 Anweisung MC\_MoveJog**

Tabelle 9- 41 Anweisung MC\_MoveJog

KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveJog_DB" (   Axis:=_multi_fb_in_,   JogForward:=_bool_in_,   JogBackward:=_bool_in_,   Velocity:=_real_in_,   InVelocity=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);</pre>	<p>Mit der Anweisung MC_MoveJog bewegen Sie die Achse konstant mit der angegebenen Geschwindigkeit im Tipbetrieb. Diese Anweisung dient üblicherweise zu Test- und Inbetriebnahmezwecken.</p> <p>Um die Anweisung MC_MoveJog zu verwenden, muss die Achse zunächst freigegeben werden.</p>

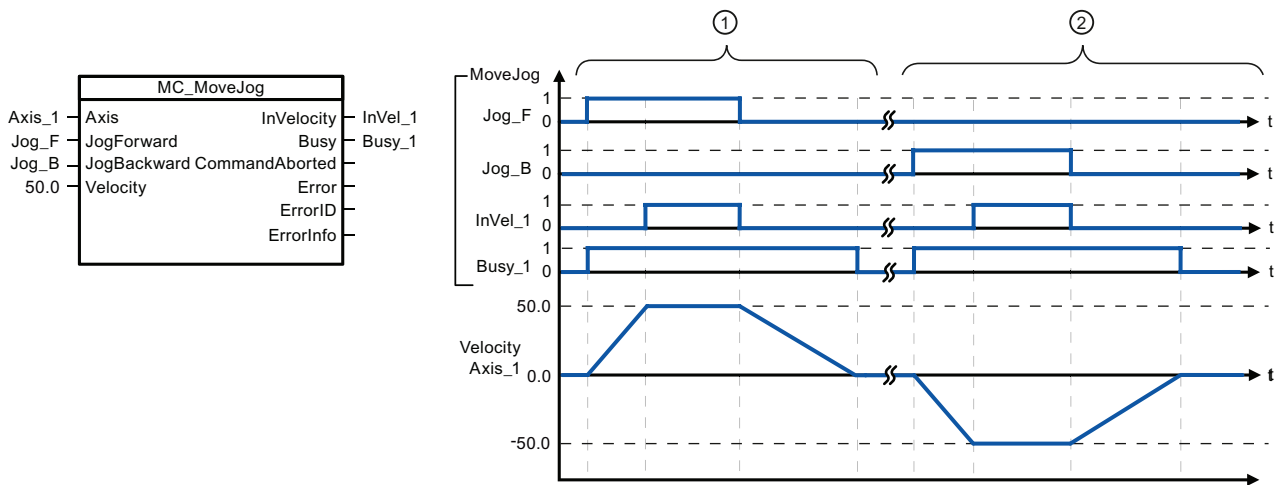
- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_MoveJog\_DB" der Name des Instanz-DBs.

Tabelle 9- 42 Parameter für die MC\_MoveJog-Anweisung

Parameter und Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1 Technologieobjekt "Achse"
JogForward <sup>1</sup>	IN	Bool Solange der Parameter WAHR ist, bewegt sich die Achse mit der im Parameter "Velocity" angegebenen Geschwindigkeit in positiver Richtung. Das Vorzeichen des Werts im Parameter "Velocity" wird ignoriert. (Standardwert: Falsch)

Parameter und Datentyp	Datentyp	Beschreibung
JogBackward <sup>1</sup>	IN	Bool
Velocity	IN	Real
InVelocity	OUT	Bool
Busy	OUT	Bool
CommandAborted	OUT	Bool
Error	OUT	Bool
ErrorID	OUT	Word
ErrorInfo	OUT	Word

<sup>1</sup> Wenn beide Parameter JogForward und JogBackward gleichzeitig WAHR sind, stoppt die Achse mit der konfigurierten Verzögerung. Ein Fehler wird in den Parametern "Error", "ErrorID" und "ErrorInfo" gemeldet.



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 5,0

- ① Die Achse wird im Tippbetrieb über "Jog\_F" in positiver Richtung bewegt. Wenn die Zielgeschwindigkeit 50,0 erreicht ist, wird dies über "InVelo\_1" gemeldet. Die Achse bremsst erneut bis zum Stillstand, nachdem Jog\_F zurückgesetzt wird.
- ② Die Achse wird im Tippbetrieb über "Jog\_B" in negativer Richtung bewegt. Wenn die Zielgeschwindigkeit 50,0 erreicht ist, wird dies über "InVelo\_1" gemeldet. Die Achse bremsst erneut bis zum Stillstand, nachdem Jog\_B zurückgesetzt wird.

### Übersteuerungsantwort

Die Aufgabe MC\_MoveJog kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

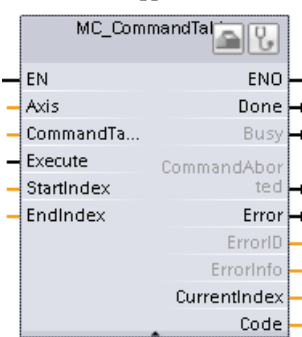
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

Die neue Aufgabe MC\_MoveJog bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog

### 9.3.3.9 Anweisung MC\_CommandTable

Tabelle 9- 43 Anweisung MC\_CommandTable

KOP/FUP	SCL	Beschreibung
	<pre>"MC_CommandTable_DB" (   Axis:= _multi_fb_in_,   CommandTable:= _multi_fb_in_,   Execute:= _bool_in_,   StartIndex:= _uint_in_,   EndIndex:= _uint_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _word_out_,   CurrentIndex=&gt; _uint_out_,   Code=&gt; _word_out_ );</pre>	<p>Führt eine Reihe einzelner Bewegungen für eine Motorsteuerungsachse aus, die zu einer Bewegungsfolge verbunden werden können.</p> <p>Einzelne Bewegungen werden in einer Befehlstabelle eines Technologieobjekts für die Impulsfolge konfiguriert (TO_CommandTable_PTO).</p>

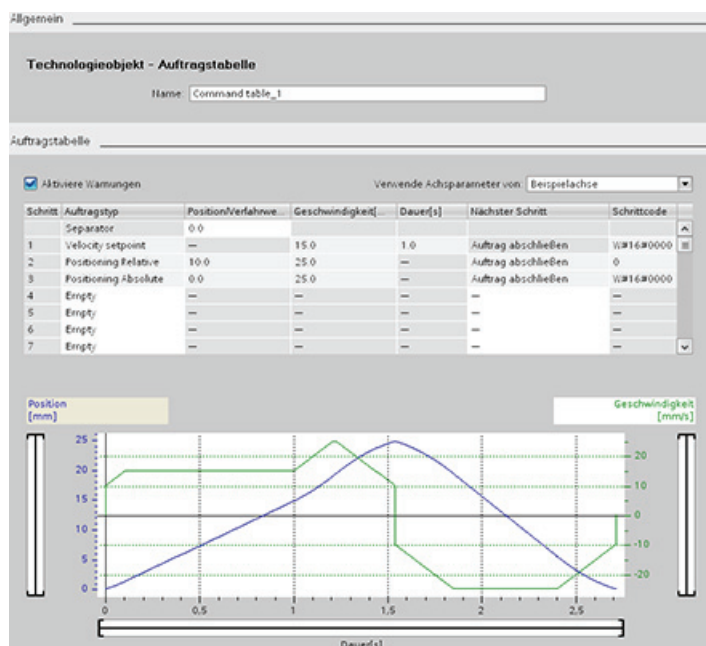
- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_CommandTable\_DB" der Name des Instanz-DBs.

Tabelle 9- 44 Parameter für die MC\_CommandTable-Anweisung

Parameter und Datentyp	Datentyp	Anfangswert	Beschreibung	
Axis	IN	TO_Axis_1	-	Technologieobjekt "Achse"
Table	IN	TO_CommandTable_1	-	Technologieobjekt "Befehlstabelle"
Execute	IN	Booll	FALSCH	Auftrag bei steigender Flanke starten
StartIndex	IN	Int	1	Verarbeitung der Befehlstabelle mit diesem Schritt starten Grenzwerte: $1 \leq \text{StartIndex} \leq \text{EndIndex}$

Parameter und Datentyp	Datentyp	Datentyp	Anfangswert	Beschreibung
EndIndex	IN	Int	32	Verarbeitung der Befehlstabelle mit diesem Schritt beenden Grenzwerte: StartIndex ≤ <b>EndIndex</b> ≤ 32
Done	OUT	Bool	FALSCH	Verarbeitung von MC_CommandTable erfolgreich ausgeführt
Busy	OUT	Bool	FALSCH	Operation in Bearbeitung
CommandAborted	OUT	Bool	FALSCH	Die Aufgabe wurde während der Bearbeitung von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	FALSCH	Ein Fehler ist während der Bearbeitung aufgetreten. Die Ursache wird von den Parametern ErrorID und ErrorInfo. angegeben
ErrorID	OUT	Word	16#0000	Fehlerkennung
ErrorInfo	OUT	Word	16#0000	Fehlerinformation
Step	OUT	Int	0	Schritt wird gerade bearbeitet
Code	OUT	Word	16#0000	Anwenderdefinierte Kennung des in Bearbeitung befindlichen Schritts

Die gewünschte Bewegungsfolge können Sie im Konfigurationsfenster "Befehlstabelle" erstellen und das Ergebnis in der grafischen Darstellung im Kurvendiagramm prüfen.



Sie können die Befehlstypen auswählen, die für die Verarbeitung der Befehlstabelle verwendet werden sollen. Bis zu 32 Aufträge können eingegeben werden. Die Befehle werden der Reihe nach verarbeitet.



Tabelle 9- 45 MC\_CommandTable-Befehlstypen

<b>Befehlstyp</b>	<b>Beschreibung</b>
Empty	Dieser Befehl dient als Platzhalter für jeden hinzuzufügenden Befehl. Der leere Eintrag wird bei der Verarbeitung der Befehlstabelle ignoriert.
Halt	Der Befehl hält die Achse an. Hinweis: Der Befehl wird nur nach einem Befehl "Velocity setpoint" durchgeführt.
Positioning Relative	Der Befehl positioniert die Achse basierend auf der Distanz. Er bewegt die Achse um die angegebene Distanz und Geschwindigkeit.
Positioning Absolute	Der Befehl positioniert die Achse basierend auf der Lage. Er bewegt die Achse mit der vorgegebenen Geschwindigkeit an die angegebene Position.
Velocity setpoint	Der Befehl bewegt die Achse mit der angegebenen Geschwindigkeit.
Wait	Der Befehl wartet, bis der angegebene Zeitraum abgelaufen ist. "Wait" stoppt keine aktive Verfahrbewegung.
Separator	Der Befehl fügt eine Trennlinie ("Separator") oberhalb der ausgewählten Linie ein. Die Trennlinie ermöglicht die Definition mehrerer Profile in einer Befehlstabelle.

Voraussetzungen für die Ausführung von MC\_CommandTable:

- Das Technologieobjekt TO\_Axis\_PTO V2.0 muss ordnungsgemäß konfiguriert sein.
- Das Technologieobjekt TO\_CommandTable\_PTO muss korrekt konfiguriert sein.
- Die Achse muss freigegeben sein.

#### **Übersteuerungsantwort**

Die Aufgabe MC\_CommandTable kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

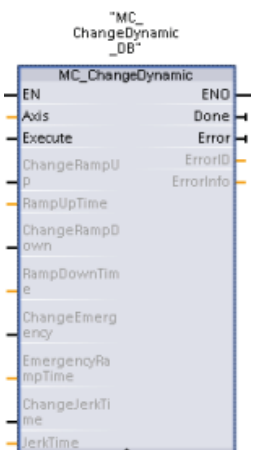
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog
- MC\_CommandTable

Die neue Aufgabe MC\_CommandTable bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog
- MC\_CommandTable
- Der aktuelle Bewegungssteuerungsauftrag mit dem Start des ersten Befehls "Positioning Relative", "Positioning Absolute", "Velocity setpoint" oder "Halt"

### 9.3.3.10 MC\_ChangeDynamic

Tabelle 9- 46 Anweisung MC\_ChangeDynamic

KOP/FUP	SCL	Beschreibung
	<pre>"MC_ChangeDynamic_DB" (     Execute:=_bool_in_,     ChangeRampUp:=_bool_in_,     RampUpTime:=_real_in_,     ChangeRampDown:=_bool_in_,     RampDownTime:=_real_in_,     ChangeEmergency:=_bool_in_,     EmergencyRampTime:=_real_in_,     ChangeJerkTime:=_bool_in_,     JerkTime:=_real_in_,     Done=&gt;_bool_out_,     Error=&gt;_bool_out_,     ErrorID=&gt;_word_out_,     ErrorInfo=&gt;_word_out_);</pre>	<p>Ändert die Dynamikeinstellungen einer Bewegungssteuerungsachse:</p> <ul style="list-style-type: none"> <li>• Wert der Hochlaufzeit (Beschleunigung) ändern</li> <li>• Wert der Rücklaufzeit (Verzögerung) ändern</li> <li>• Wert der Not-Aus-Rücklaufzeit (Not-Aus-Verzögerung) ändern</li> <li>• Wert der Glättungszeit (Ruck) ändern</li> </ul>

- 1 STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC\_ChangeDynamic\_DB" der Name des Instanz-DBs.

Tabelle 9- 47 Parameter für die MC\_ChangeDynamic-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten des Befehls bei einer positiven Flanke. Standardwert: FALSE
ChangeRampUp	IN	Bool	TRUE = Hochlaufzeit in Übereinstimmung mit dem Eingangsparameter "RampUpTime" ändern. Standardwert: FALSE
RampUpTime	IN	Real	Zeit (in Sekunden), während der ohne Ruckbegrenzung vom Stillstand zur konfigurierten Maximalgeschwindigkeit beschleunigt werden soll. Standardwert: 5.00 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.Acceleration. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variable gezeigt.
ChangeRampDown	IN	Bool	TRUE = Rücklaufzeit in Übereinstimmung mit dem Eingangsparameter "RampDownTime" ändern. Standardwert: FALSE
RampDownTime	IN	Real	Zeit (in Sekunden), während der die Achse ohne Ruckbegrenzung von der konfigurierten Maximalgeschwindigkeit bis zum Stillstand verzögert werden soll. Standardwert: 5.00 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.Deceleration. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variable gezeigt.

Parameter und Datentyp		Datentyp	Beschreibung
ChangeEmergency	IN	Bool	TRUE = Not-Aus-Rücklaufzeit in Übereinstimmung mit dem Eingangsparameter "EmergencyRampTime" ändern. Standardwert: FALSE
EmergencyRampTime	IN	Real	Zeit (in Sekunden), während der die Achse im Not-Aus-Modus ohne Ruckbegrenzung von der konfigurierten Maximalgeschwindigkeit bis zum Stillstand verzögert werden soll. Standardwert: 2.00 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.EmergencyDeceleration. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variable gezeigt.
ChangeJerkTime	IN	Bool	TRUE = Glättungszeit in Übereinstimmung mit dem Eingangsparameter "JerkTime" ändern. Standardwert: FALSE
JerkTime	IN	Real	Glättungszeit (in Sekunden) für die Beschleunigungs- und Verzögerungsrampen der Achse. Standardwert: 0.25 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.Jerk. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variable gezeigt.
Done	OUT	Bool	TRUE = Die geänderten Werte wurden in den Technologiedatenbaustein geschrieben. Die Beschreibung der Variablen wird angezeigt, wenn die Änderung wirksam wird. Standardwert: FALSE
Error	OUT	Bool	TRUE = Während der Ausführung des Befehls trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt. Standardwert: FALSE
ErrorID	OUT	Word	Fehlerkennung. Standardwert: 16#0000
ErrorInfo	IN	Word	Fehlerinformationen. Standardwert: 16#0000

Voraussetzungen für die Ausführung von MC\_ChangeDynamic:

- Das Technologieobjekt TO\_Axis\_PTO V2.0 muss ordnungsgemäß konfiguriert sein.
- Die Achse muss freigegeben sein.

## Übersteuerungsantwort

Ein Befehl MC\_ChangeDynamic kann von keinem anderen Bewegungssteuerungsbefehl abgebrochen werden.

Der neue Befehl MC\_ChangeDynamic bricht keine aktiven Bewegungssteuerungsaufgaben ab.

---

### Hinweis

Für die Eingangsparameter "RampUpTime", "RampDownTime", "EmergencyRampTime" und "RoundingOffTime" können Werte angegeben werden, die dazu führen, dass die resultierenden Achsenparameter "Beschleunigung", "Verzögerung", "Not-Aus-Verzögerung" und "Ruck" außerhalb der zulässigen Grenzwerte liegen.

Stellen Sie sicher, dass Sie die Parameter MC\_ChangeDynamic innerhalb der Grenzwerte der Dynamikkonfiguration des Technologieobjekts "Achse" halten.

---

### 9.3.4 Funktionsweise der Bewegungssteuerung bei der S7-1200

#### 9.3.4.1 Für die Bewegungssteuerung verwendete CPU-Ausgänge

Die CPU bietet vier Impulsgeneratoren. Jeder Impulsgenerator bietet einen Impulsausgang und einen Richtungsausgang zum Steuern eines Schrittmotorantriebs oder eines Servomotorantriebs mit Impulsschnittstelle. Der Impulsausgang bietet den Antrieb mit den für die Motorbewegung erforderlichen Impulsen. Der Richtungsausgang steuert die Verfahrrichtung des Antriebs.

Impuls- und Richtungsausgänge sind einander dauerhaft zugewiesen. Integrierte CPU-Ausgänge und Ausgänge eines Signalboards können als Impuls- und Richtungsausgänge verwendet werden. Sie wählen zwischen integrierten CPU-Ausgängen und Ausgängen des Signalboards während der Gerätekonfiguration der Impulsgeneratoren (PTO/PWM) im Register "Eigenschaften". Nur PTO (Pulse Train Output) gilt für die Bewegungssteuerung.

Der PTO-Ausgang erzeugt einen Rechteckausgang mit variabler Frequenz. Die Impulserzeugung wird über Konfigurations- und Ausführungsinformationen gesteuert, die über die H/W-Konfiguration und/oder SFCs/SFBs vorgegeben werden.

Während sich die CPU im Betriebszustand RUN befindet, steuern je nach Auswahl des Anwenders entweder die Werte im Prozessabbild oder die Ausgänge des Impulsgenerators die digitalen Ausgänge. Im Betriebszustand STOP steuert der PTO-Generator die Ausgänge nicht.

Tabelle 9- 48 Adresszuweisungen für die Impuls- und Richtungsausgänge

Verwendung von Ausgängen für die Bewegungssteuerung		
	Impuls	Richtung
PTO 0		
Integrierte E/A	A0.0	A0.1
SB-E/A	A4.0	A4.1
PTO 1		
Integrierte E/A	A0.2	A0.3
SB-E/A	A4.2	A4.3
PTO 2		
Integrierte E/A	A0.4 <sup>1</sup>	A0.5 <sup>1</sup>
SB-E/A	A4.0	A4.1
PTO 3		
Integrierte E/A	A0.6 <sup>2</sup>	A0.7 <sup>2</sup>
SB-E/A	A4.2	A4.3

<sup>1</sup> Die CPU 1211C hat keine Ausgänge A0.4, A0.5, A0.6 und A0.7. Deshalb können diese Ausgänge bei der CPU 1211C nicht verwendet werden.

<sup>2</sup> Die CPU 1212C hat keine Ausgänge A0.6 und A0.7. Deshalb können diese Ausgänge bei der CPU 1212C nicht verwendet werden.

<sup>3</sup> Diese Tabelle gilt für die PTO-Funktionen der CPU 1211C, CPU 1212C, CPU 1214C und CPU 1215C.

## Antriebsschnittstelle

Für die Bewegungssteuerung können Sie optional für eine Antriebsschnittstelle die Zustände "Antrieb freigegeben" und "Antrieb bereit" konfigurieren. Wenn Sie die Antriebsschnittstelle verwenden, können der Digitalausgang für "Antrieb freigegeben" und der Digitaleingang für "Antrieb bereit" frei ausgewählt werden.

### Hinweis

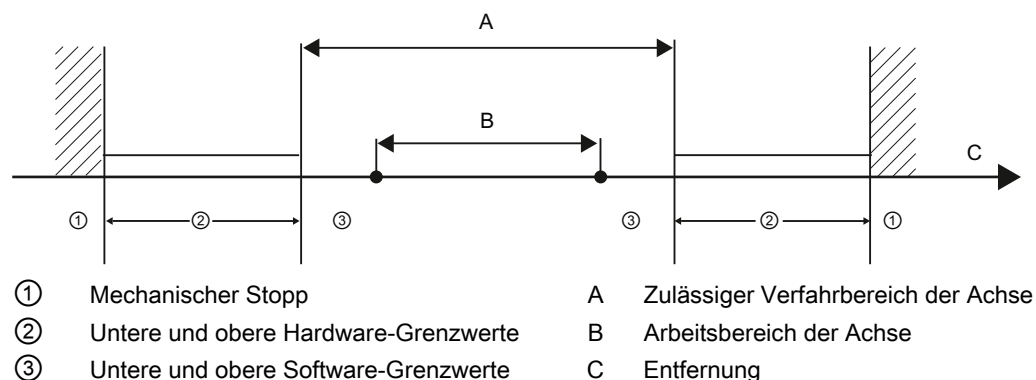
Die Firmware übernimmt die Steuerung über die entsprechenden Impuls- und Richtungsausgänge, wenn die PTO (Pulse Train Output) ausgewählt und einer Achse zugewiesen wurde.

Mit dieser Übernahme der Steuerungsfunktion wird auch die Verbindung zwischen dem Prozessabbild und dem E/A-Ausgang getrennt. Der Anwender hat zwar die Möglichkeit, über das Anwenderprogramm oder die Beobachtungstabelle in das Prozessabbild der Impuls- und Richtungsausgänge zu schreiben, diese Daten werden jedoch nicht an die Ausgänge übertragen. Entsprechend ist es auch nicht möglich, die Ausgänge über das Anwenderprogramm oder die Beobachtungstabelle zu überwachen. Die ausgelesenen Informationen geben lediglich den Wert des Prozessabbilds wieder und entsprechen in keiner Weise dem tatsächlichen Zustand der Ausgänge.

Bei allen anderen CPU-Ausgängen, die nicht permanent von der CPU-Firmware verwendet werden, kann der Zustand der Ausgänge wie üblich über das Prozessabbild gesteuert und überwacht werden.

### 9.3.4.2 Hardware- und Software-Endschalter für die Bewegungssteuerung

Mit den Hardware- und Software-Endschaltern begrenzen Sie den zulässigen Verfahrbereich und den Arbeitsbereich Ihrer Achse.



Hardware- und Software-Endschalter müssen vor der Verwendung in der Konfiguration oder im Anwenderprogramm aktiviert werden. Software-Endschalter sind erst nach der Referenzpunktfahrt der Achse aktiv.

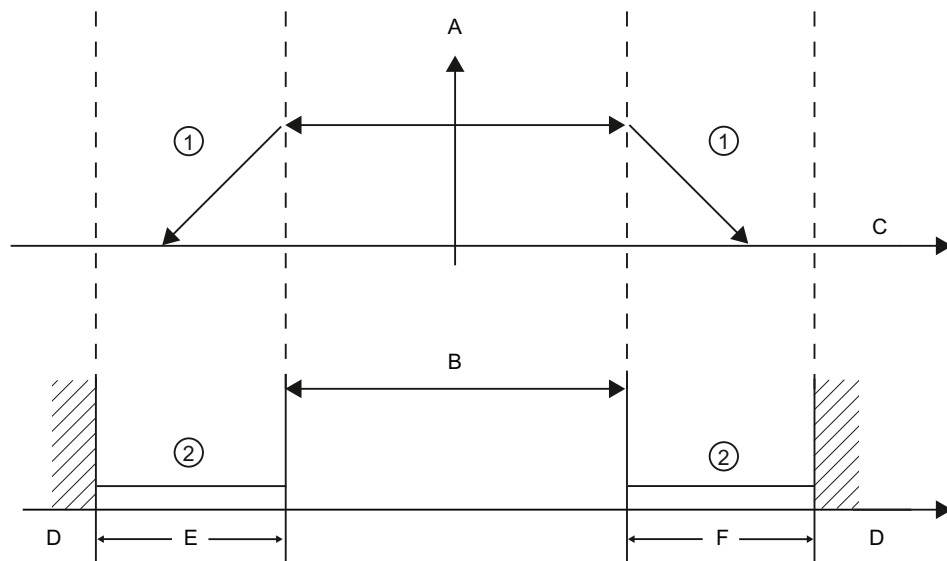
### Hardware-Endschalter

Hardware-Endschalter geben den maximalen Verfahrbereich der Achse vor. Hardware-Endschalter sind physikalische Schaltelemente, die an alarmfähige Eingänge der CPU angeschlossen werden müssen. Verwenden Sie nur Hardware-Endschalter, die nach der Anfahrt dauerhaft geschaltet bleiben. Der Schaltzustand kann erst nach der Rückkehr in den zulässigen Verfahrbereich rückgängig gemacht werden.


Tabelle 9- 49 Verfügbare Eingänge für Impulsgeneratoren

Beschreibung	RPS	LIM-	LIM+
PTO 0			
Integrierte E/A		E0.0 - E1.5	
SB-E/A		E4.0 - E4.3	
PTO 1			
Integrierte E/A		E0.0 - E1.5	
SB-E/A		E4.0 - E4.3	
PTO 2			
Integrierte E/A		E0.0 - E1.5	
SB-E/A		E4.0 - E4.3	
PTO 3			
Integrierte E/A		E0.0 - E1.5	
SB-E/A		E4.0 - E4.3	

Wenn die Hardware-Endschalter angefahren werden, bremst die Achse mit der konfigurierten Notfallverzögerung bis zum Stillstand. Die angegebene Notfallverzögerung muss ausreichend sein, um die Achse vor dem mechanischen Stopp zuverlässig zu stoppen. Das folgende Diagramm zeigt das Verhalten der Achse nach Anfahrt der Hardware-Endschalter.



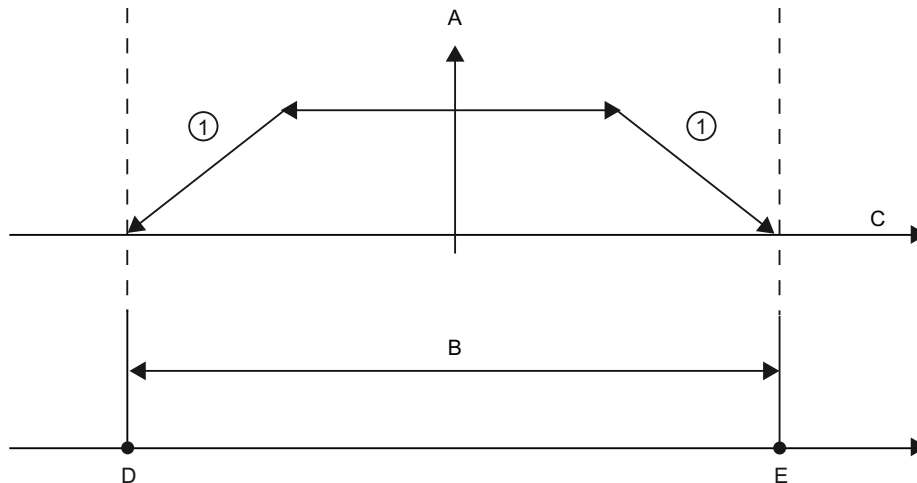
- ① Die Achse brems mit der konfigurierten Notfallverzögerung bis zum Stillstand.
- ② Bereich, in dem die Hardware-Endschalter den Zustand "angefahren" signalisieren.
- A [Geschwindigkeit]
- B Zulässiger Verfahrbereich
- C Entfernung
- D Mechanischer Stopp
- E Unterer Hardware-Endschalter
- F Oberer Hardware-Endschalter

 <b>WARNUNG</b>
<p>Wenn die bisherige Einstellung der Filterzeit eines Digitaleingangskanals verändert wird, muss möglicherweise bis zu 20,0 ms lang ein neuer Eingangswert mit Pegel "0" vorhanden sein, damit der Filter vollständig auf neue Eingänge reagiert. Während dieses Zeitraums werden kurze Impulse mit Pegel "0", die kürzer als 20,0 ms sind, möglicherweise nicht erkannt oder gezählt.</p> <p>Das Verändern der Filterzeiten kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.</p> <p>Um sicherzustellen, dass eine neue Filterzeit sofort wirksam wird, müssen Sie die CPU aus- und wieder einschalten.</p>

## Software-Endschalter

Software-Endschalter begrenzen den Arbeitsbereich der Achse. Sie sollten sich relativ zum Verfahrbereich innerhalb der Hardware-Endschalter befinden. Weil die Positionen der Software-Endschalter flexibel eingerichtet werden können, lässt sich der Arbeitsbereich der Achse individuell je nach aktuellem Verfahrprofil einschränken. Im Gegensatz zu Hardware-Endschaltern werden Software-Endschalter ausschließlich über die Software implementiert und benötigen keine eigenen Schaltelemente.

Wenn Software-Endschalter aktiviert werden, wird eine aktive Bewegung an der Position des Software-Endschalters gestoppt. Die Achse wird mit der konfigurierten Verzögerung gebremst. Das folgende Diagramm zeigt das Verhalten der Achse bis zum Erreichen des Software-Endschalters.



- ① Die Achse bremst mit der konfigurierten Verzögerung bis zum Stillstand.
- A [Geschwindigkeit]
- B Arbeitsbereich
- C Entfernung
- D Unterer Software-Endschalter
- E Oberer Software-Endschalter

Verwenden Sie zusätzlich Hardware-Endschalter, wenn sich ein mechanischer Endstopp nach den Software-Endschaltern befindet und die Gefahr mechanischer Schäden besteht.

### Weitere Informationen

Ihr Anwenderprogramm kann die Grenzwerte der Hardware- oder Softwareposition durch Aktivieren oder Deaktivieren der Hardware- und Software-Grenzwertfunktionen übersteuern. Die Auswahl erfolgt über den Achsen-DB.

- Um die Hardware-Grenzwertfunktion zu aktivieren, rufen Sie die Variable "Active" (Bool) im DB-Pfad "<Achsename>/Config/PositionLimits\_HW" auf. Der Zustand der Variable "Active" aktiviert oder deaktiviert die Verwendung von Hardware-Positionsgrenzwerten.
- Um die Software-Grenzwertfunktion zu aktivieren, rufen Sie die Variable "Active" (Bool) im DB-Pfad "<Achsename>/Config/PositionLimits\_SW" auf. Der Zustand der Variable "Active" aktiviert oder deaktiviert die Verwendung von Software-Positionsgrenzwerten.

Sie können die Software-Positionsgrenzwerte auch mit Ihrem Anwenderprogramm ändern (beispielsweise um die Maschineneinrichtung flexibler zu gestalten oder um die Umrüstzeiten von Maschinen zu verkürzen). Ihr Anwenderprogramm kann neue Werte in die Variablen "MinPosition" und "MaxPosition" (physikalische Einheiten im Format Real) im DB "<Achsename>/Config/PositionLimits\_SW" schreiben.



### 9.3.4.3 Referenzpunktfahrt

Bei der Referenzpunktfahrt werden die Achsenkoordinaten an die reale, physikalische Position des Antriebs angepasst. (Befindet sich der Antrieb gegenwärtig an Position x, wird die Achse in Position x gebracht.) Bei positionsgesteuerten Achsen beziehen sich die Einträge und Anzeigen für die Position exakt auf diese Achsenkoordinaten.

---

#### Hinweis

Die Übereinstimmung zwischen den Achsenkoordinaten und der realen Situation ist äußerst wichtig. Dieser Schritt ist erforderlich, um sicherzustellen, dass die absolute Zielposition der Achse auch exakt mit dem Antrieb erreicht wird.

---

Die Anweisung MC\_Home löst die Referenzpunktfahrt der Achse aus.

Es gibt vier verschiedene Funktionen für die Referenzpunktfahrt. Die ersten beiden Funktionen ermöglichen es dem Anwender, die aktuelle Position der Achse einzustellen, und die beiden zweiten positionieren die Achse in Bezug auf einen Referenzpunktsensor.

- Betriebsart 0 - Direkte Referenzpunktfahrt absolut: Wenn diese Betriebsart ausgeführt wird, wird der Achse genau mitgeteilt, wo sie sich befindet. Die interne Positionsvariable wird auf den Wert des Positionseingangs der Anweisung für die Referenzpunktfahrt gesetzt. Dies wird bei der Kalibrierung und Einrichtung von Maschinen verwendet.

Die Achsenposition wird unabhängig vom Referenzpunktschalter gesetzt. Aktive Verfahrbewegungen werden nicht abgebrochen. Der Wert des Eingangsparameters Position der Anweisung MC\_Home wird sofort als Referenzpunkt der Achse gesetzt. Um den Referenzpunkt einer genauen mechanischen Position zuzuweisen, muss sich die Achse zum Zeitpunkt der Referenzpunkteinstellung an dieser Position im Stillstand befinden.

- Betriebsart 1 - Direkte Referenzpunktfahrt relativ: Diese Betriebsart nutzt bei der Ausführung die interne Positionsvariable und fügt den Wert des Positionseingangs der Anweisung für die Referenzpunktfahrt in diese Variable ein. Dies wird typischerweise genutzt, um den Maschinenversatz zu berücksichtigen.

Die Achsenposition wird unabhängig vom Referenzpunktschalter gesetzt. Aktive Verfahrbewegungen werden nicht abgebrochen. Die folgende Aussage gilt für die Achsenposition nach der Referenzpunktfahrt: Neue Achsenposition = aktuelle Achsenposition + Wert des Parameters Position der Anweisung MC\_Home.

- Betriebsart 2 - Passive Referenzpunktfahrt: Wenn sich die Achse bewegt und den Referenzpunktschalter überfährt, wird die aktuelle Position als Referenzpunkt gesetzt. Bei dieser Funktion wird der normale Maschinenverschleiß und das Zahnflankenspiel berücksichtigt, um den Bedarf an manuellem Verschleißausgleich zu verhindern. Der Positionseingang der Anweisung für die Referenzpunktfahrt wird wie zuvor zur vom Referenzpunktschalter angegebenen Position addiert, um den Referenzpunkt mühelos zu versetzen.

Während der passiven Referenzpunktfahrt führt die Anweisung MC\_Home keine Referenzpunktfahrtbewegung durch. Die für diesen Schritt erforderliche Verfahrbewegung müssen Sie über andere Bewegungssteuerungsanweisungen implementieren. Wenn der Referenzpunktschalter erkannt wird, wird die Achse entsprechend der Konfiguration an den Referenzpunkt gefahren. Aktive Verfahrbewegungen werden beim Start der passiven Referenzpunktfahrt nicht abgebrochen.

- Betriebsart 3 - Aktive Referenzpunktfahrt: Diese Betriebsart ist das präziseste Verfahren für die Referenzpunktfahrt der Achse. Die anfängliche Richtung und Geschwindigkeit der Bewegung werden in den erweiterten Parametern in der Konfiguration des Technologieobjekts unter Referenzpunktfahrt konfiguriert. Dies ist abhängig von der Maschinenkonfiguration. Zudem kann festgelegt werden, ob die steigende oder fallende Flanke des Signals des Referenzpunktschalters der Referenzpunkt ist. Praktisch alle Sensoren haben einen aktiven Bereich, und wenn die Position "Steady State On" als Referenzpunktsignal verwendet wurde, besteht die Möglichkeit eines Fehlers beim Referenzpunkt, weil der aktive Bereich des EIN-Signals einen Entfernungsbereich abdeckt. Durch Verwendung der steigenden oder fallenden Flanke dieses Signals resultiert ein sehr viel präziserer Referenzpunkt. Wie bei allen anderen Betriebsarten wird der Wert des Positionseingangs der Anweisung für die Referenzpunktfahrt zum Hardware-Referenzpunkt addiert.

Bei der aktiven Referenzpunktfahrt führt die Anweisung MC\_Home die erforderliche Referenzpunktfahrt durch. Wenn der Referenzpunktschalter erkannt wird, wird die Achse entsprechend der Konfiguration an den Referenzpunkt gefahren. Aktive Verfahrbewegungen werden abgebrochen.

Bei den Betriebsarten 0 und 1 muss die Achse überhaupt nicht bewegt werden. Sie werden typischerweise bei der Einrichtung und Kalibrierung verwendet. Bei den Betriebsarten 2 und 3 muss die Achse bewegt werden und einen Sensor überfahren, der im Technologieobjekt "Achse" als Referenzpunktschalter konfiguriert ist. Der Referenzpunktschalter kann im Arbeitsbereich der Achse oder außerhalb des normalen Arbeitsbereichs der Achse, jedoch innerhalb des Bewegungsbereichs platziert werden.

### Konfiguration der Parameter für die Referenzpunktfahrt

Sie konfigurieren die Parameter für aktive und passive Referenzpunktfahrt im Konfigurationsfenster "Referenzpunktfahrt". Das Verfahren der Referenzpunktfahrt wird über den Eingangsparameter "Mode" der Bewegungssteuerungsanweisung festgelegt. Hier bedeutet Mode = 2 passive Referenzpunktfahrt und Mode = 3 bedeutet aktive Referenzpunktfahrt.

<b>ACHTUNG</b>
<p>Stellen Sie anhand einer der folgenden Maßnahmen sicher, dass die Maschine bei Richtungsumkehr nicht zu einem mechanischen Endstopp fährt:</p> <ul style="list-style-type: none"> <li>• Halten Sie die Anfahrsgeschwindigkeit gering</li> <li>• Vergrößern Sie die konfigurierte Beschleunigung/Verzögerung</li> <li>• Vergrößern Sie die Entfernung zwischen Hardware-Endschalter und mechanischem Stopp</li> </ul>

Tabelle 9- 50 Konfigurationsparameter für die Referenzpunktfahrt der Achse

Parameter	Beschreibung
Eingang Referenzpunktschalter (Aktive und passive Referenzpunktfahrt)	<p>Wählen Sie den digitalen Eingang für den Referenzpunktschalter in der Klappliste aus. Der Eingang muss alarmfähig sein. Die integrierten Eingänge der CPU und die Eingänge eines gesteckten Signalboards können als Eingänge für den Referenzpunktschalter ausgewählt werden.</p> <p>Die Standardfilterzeit für die digitalen Eingänge beträgt 6,4 ms. Wenn die digitalen Eingänge als Referenzpunktschalter verwendet werden, kann dies zu unerwünschten Verzögerungen und damit zu Ungenauigkeiten führen. Abhängig von der verringerten Geschwindigkeit und dem Ausmaß des Referenzpunktschalters wird der Referenzpunkt möglicherweise nicht erkannt. Die Filterzeit kann in der Gerätekonfiguration der digitalen Eingänge unter "Eingangsfiler" eingestellt werden.</p> <p>Die angegebene Filterzeit muss kleiner als die Dauer des Eingangssignals am Referenzpunktschalter sein.</p>
Automatische Umkehr nach Erreichen der Hardware-Endschalter (Nur bei aktiver Referenzpunktfahrt)	<p>Aktivieren Sie das Kontrollkästchen, um den Hardware-Endschalter als Umkehrnocken für die Referenzpunktfahrt zu nutzen. Die Hardware-Endschalter müssen für die Richtungsumkehr konfiguriert und aktiviert sein.</p> <p>Wenn der Hardware-Endschalter während der aktiven Referenzpunktfahrt erreicht wird, bremst die Achse mit der konfigurierten Verzögerung (nicht mit der Notfallverzögerung) und kehrt die Richtung um. Der Referenzpunktschalter wird dann in umgekehrter Richtung erfasst.</p> <p>Wenn die Richtungsumkehr nicht aktiv ist und die Achse den Hardware-Endschalter während der aktiven Referenzpunktfahrt erreicht, wird die Referenzpunktfahrt mit einem Fehler abgebrochen und die Achse mit der Notfallverzögerung gebremst.</p>
Anfahrriichtung (Aktive und passive Referenzpunktfahrt)	<p>Bei der Richtungsauwahl legen Sie die "Anfahrriichtung" für die aktive Referenzpunktfahrt zum Suchen des Referenzpunktschalters sowie die Richtung der Referenzpunktfahrt fest. Die Richtung der Referenzpunktfahrt legt die Verfahrrichtung der Achse zum Anfahren der konfigurierten Seite des Referenzpunktschalters fest, um die Referenzpunktfahrt durchzuführen.</p>

Parameter	Beschreibung
Referenzpunktschalter (Aktive und passive Referenzpunktfahrt)	<ul style="list-style-type: none"> <li>• Aktive Referenzpunktfahrt: Geben Sie an, ob die Achse auf der linken oder rechten Seite des Referenzpunktschalters referenziert werden soll. Je nach Startposition der Achse und der Konfiguration der Parameter für die Referenzpunktfahrt kann sich die Abfolge der Referenzpunktfahrt vom Diagramm im Konfigurationsfenster unterscheiden.</li> <li>• Passive Referenzpunktfahrt: Bei der passiven Referenzpunktfahrt müssen die Verfahrbewegungen zum Zweck der Referenzpunktfahrt vom Anwender über Bewegungssteuerungsbefehle implementiert werden. Die Seite des Referenzpunktschalters, auf der die Referenzpunktfahrt erfolgt, ist von den folgenden Faktoren abhängig: <ul style="list-style-type: none"> <li>– Konfigurierte Anfahrriichtung</li> <li>– Konfigurierter Referenzpunktschalter</li> <li>– Aktuelle Verfahrriichtung während der passiven Referenzpunktfahrt</li> </ul> </li> </ul>
Anfahrsgeschwindigkeit (Nur bei aktiver Referenzpunktfahrt)	<p>Geben Sie die Geschwindigkeit an, mit der der Referenzpunktschalter während der Referenzpunktfahrt gesucht werden soll.</p> <p>Grenzwerte (unabhängig von der ausgewählten Anwendereinheit):  Start-/Stoppgeschwindigkeit ≤ Anfahrsgeschwindigkeit ≤ Maximalgeschwindigkeit</p>
Verringerte Geschwindigkeit (Nur bei aktiver Referenzpunktfahrt)	<p>Geben Sie die Geschwindigkeit an, mit der die Achse den Referenzpunktschalter für die Referenzpunktfahrt anfährt.</p> <p>Grenzwerte (unabhängig von der ausgewählten Anwendereinheit):  Start-/Stoppgeschwindigkeit ≤ verringerte Geschwindigkeit ≤ Maximalgeschwindigkeit</p>
Referenzpunktversatz (Nur bei aktiver Referenzpunktfahrt)	<p>Wenn der gewünschte Referenzpunkt von der Position des Referenzpunktschalters abweicht, kann der Referenzpunktversatz in diesem Feld angegeben werden.</p> <p>Ist der Wert ungleich 0, führt die Achse die folgenden Vorgänge nach der Referenzpunktfahrt am Referenzpunktschalter durch:</p> <ol style="list-style-type: none"> <li>1. Bewegen der Achse mit verringerter Geschwindigkeit um den Wert des Referenzpunktversatzes.</li> <li>2. Wenn die Position des Referenzpunkts erreicht ist, wird die Achsenposition auf den absoluten Referenzpunkt gesetzt. Der absolute Referenzpunkt wird über den Parameter "Position" der Bewegungssteuerungsanweisung "MC_Home" angegeben.</li> </ol> <p>Grenzwerte (unabhängig von der ausgewählten Anwendereinheit):  -1,0e12 ≤ Referenzpunktversatz ≤ 1,0e12</p>

Tabelle 9- 51 Die Referenzpunktfahrt beeinflussende Faktoren

Beeinflussende Faktoren:			Ergebnis:
Konfiguration Anfahrriichtung	Konfiguration Referenzpunktschalter	Aktuelle Verfahrrichtung	Referenzpunktfahrt zu Referenzpunktschalter
Positiv	"Linke (negative) Seite"	Positive Richtung	Links
		Negative Richtung	Rechts
Positiv	"Rechte (positive) Seite"	Positive Richtung	Rechts
		Negative Richtung	Links
Negativ	"Linke (negative) Seite"	Positive Richtung	Rechts
		Negative Richtung	Links
Negativ	"Rechte (positive) Seite"	Positive Richtung	Links
		Negative Richtung	Rechts

### Abfolge bei der aktiven Referenzpunktfahrt

Sie aktive Referenzpunktfahrt starten Sie mit der Bewegungssteuerungsanweisung "MC\_Home" (Eingangsparameter Mode = 3). Der Eingangsparameter "Position" gibt in diesem Fall die absoluten Koordinaten des Referenzpunkts an. Alternativ können Sie die aktive Referenzpunktfahrt zu Testzwecken im Steuerpanel starten.

Das folgende Diagramm zeigt ein Beispiel einer Kennlinie für eine aktive Referenzpunktfahrt mit den folgenden Konfigurationsparametern:

- "Anfahrriichtung" = "Positive Anfahrriichtung"
- "Referenzpunktschalter" = "Rechte (positive) Seite"
- Wert des "Referenzpunktversatzes" > 0

Tabelle 9- 52 Geschwindigkeitskennlinie für die Referenzpunktfahrt bei der Bewegungssteuerung

Funktionsweise		Hinweise	
		A	Anfahrsgeschwindigkeit
		B	Verringerte Geschwindigkeit
		C	Referenzpunktkoordinaten
		D	Referenzpunktversatz
①	Suchphase (blaues Kennliniensegment): Wenn die aktive Referenzpunktfahrt startet, beschleunigt die Achse auf die konfigurierte "Anfahrsgeschwindigkeit" und sucht bei dieser Geschwindigkeit nach dem Referenzpunktschalter.		
②	Referenzpunktfahrt (roter Kennlinienabschnitt): Wenn der Referenzpunktschalter erkannt ist, bremst die Achse in diesem Beispiel und kehrt um, um auf der konfigurierten Seite des Referenzpunktschalters bei der konfigurierten "verringerten Geschwindigkeit" an den Referenzpunkt zu fahren.		
③	Verfahren zum Referenzpunkt (grünes Kennliniensegment): Nach der Referenzpunktfahrt am Referenzpunktschalter verfährt die Achse bei "verringertem Geschwindigkeit" zu den "Referenzpunktkoordinaten". Beim Erreichen der "Referenzpunktkoordinaten" wird die Achse an dem Positionswert gestoppt, der im Eingangsparameter Position der Anweisung MC_Home angegeben wurde.		

**Hinweis**

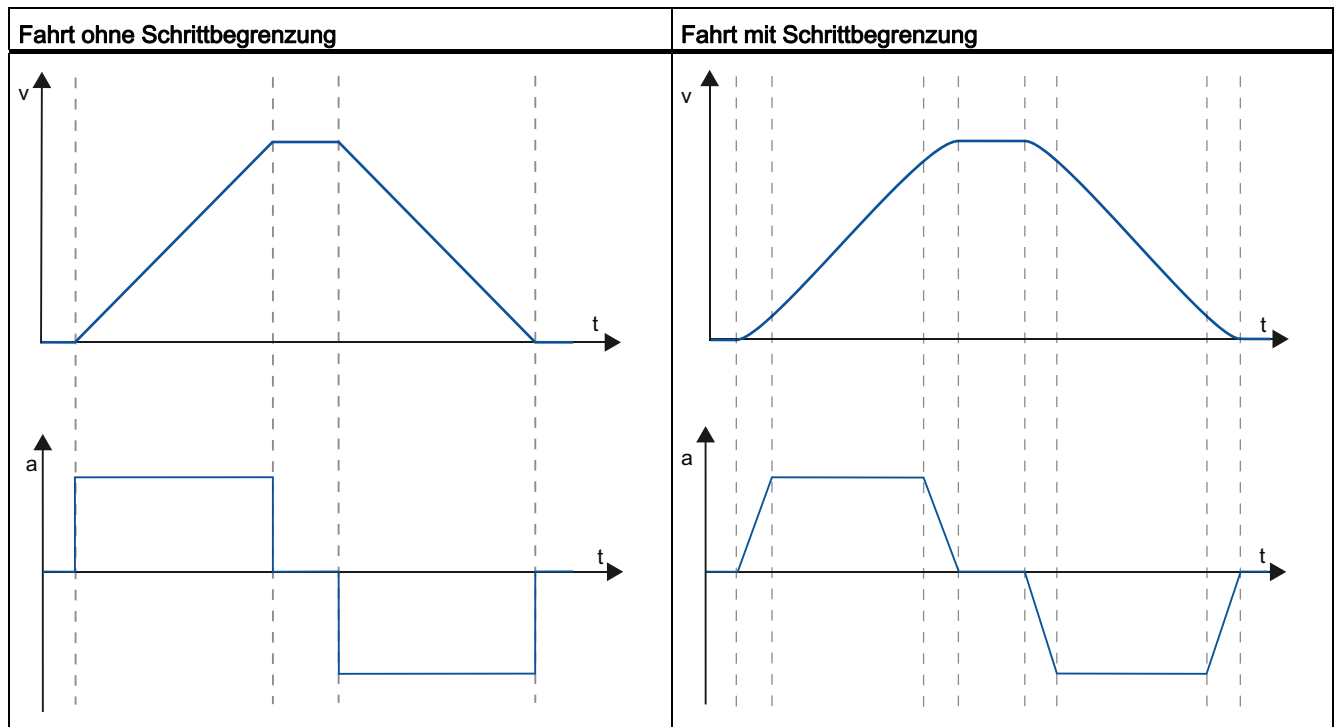
Wenn die Referenzpunktsuche nicht wie erwartet funktioniert, prüfen Sie die den Hardware-Grenzwerten oder dem Referenzpunkt zugewiesenen Eingänge. Möglicherweise wurden die Flankenalarme für diese Eingänge in der Gerätekonfiguration deaktiviert.

Prüfen Sie die Konfigurationsdaten des betroffenen Technologieobjekts "Achse", um festzustellen, welche Eingänge ggf. zu "HW Low Limit Switch Input", "HW High Limit Switch Input" und "Input reference point switch" zugewiesen sind. Öffnen Sie dann die Gerätekonfiguration der CPU und untersuchen Sie jeden der zugewiesenen Eingänge. Prüfen Sie, ob "Erkennung steigende Flanke aktivieren" und "Erkennung fallende Flanke aktivieren" beide ausgewählt sind. Sind diese Eigenschaften nicht ausgewählt, löschen Sie die angegebenen Eingänge in der Achsenkonfiguration und wählen sie erneut aus.

### 9.3.4.4 Ruckbegrenzung

Mit der Ruckbegrenzung können Sie die mechanischen Beanspruchungen während der Beschleunigungs- und Verzögerungsphase verringern. Der Wert für die Beschleunigung und Verzögerung wird nicht abrupt verändert, wenn die Schrittbegrenzung aktiv ist, der Wert wird stattdessen während einer Übergangsphase angepasst. Die folgende Abbildung zeigt die Geschwindigkeits- und Beschleunigungskurve ohne und mit Ruckbegrenzung.

Tabelle 9- 53 Ruckbegrenzung



Die Ruckbegrenzung sorgt für ein "geglättetes" Geschwindigkeitsprofil der Achsbewegung. Dadurch wird ein weiches Anfahren und Abbremsen beispielsweise eines Förderbands gewährleistet.

### 9.3.5 Inbetriebnahme

#### Diagnosefunktion "Status- und Fehlerbits"

Mit der Diagnosefunktion "Status- und Fehlerbits" können Sie die wichtigsten Status- und Fehlermeldungen der Achse überwachen. Die Diagnosefunktionsanzeige ist im Online-Modus bei aktiver Achse in der Betriebsart "Manuelle Steuerung" und "Automatiksteuerung" verfügbar.

Tabelle 9- 54 Status der Achse

Status	Beschreibung
Freigegeben	Die Achse ist freigegeben und bereit, über Bewegungssteuerungsaufgaben gesteuert zu werden. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Enable)
Referenzpunktfahrt durchgeführt	Die Achse ist am Referenzpunkt und kann absolute Positionierungsaufgaben der Bewegungssteuerungsanweisung MC_MoveAbsolute durchführen. Die Achse muss sich für die relative Referenzpunktfahrt nicht am Referenzpunkt befinden. Besondere Situationen: <ul style="list-style-type: none"> <li>• Während der aktiven Referenzpunktfahrt ist der Status FALSCH.</li> <li>• Wenn für eine Achse, die sich am Referenzpunkt befindet, eine passive Referenzpunktfahrt durchgeführt wird, wird der Status während der passiven Referenzpunktfahrt auf WAHR gesetzt.</li> </ul> (Variable des Technologieobjekts: <Achsenname>.StatusBits.HomingDone)
Fehler	Im Technologieobjekt "Achse" ist ein Fehler aufgetreten. Weitere Informationen zum Fehler sind in der Automatiksteuerung an den Parametern ErrorID und ErrorInfo der Bewegungssteuerungsanweisungen verfügbar. Im Handbetrieb zeigt das Feld "Letzter Fehler" im Steuerpanel ausführliche Informationen zur Fehlerursache an. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Error)
Steuerpanel aktiv	Der Modus "Manuelle Steuerung" wurde im Steuerpanel aktiviert. Das Steuerpanel hat die Steuerpriorität über das Technologieobjekt "Achse". Die Achse kann nicht über das Anwenderprogramm gesteuert werden. (Variable des Technologieobjekts: <Achsenname>.StatusBits.ControlPanelActive)

Tabelle 9- 55 Antriebsstatus

Status	Beschreibung
Antrieb bereit	Der Antrieb ist betriebsbereit. (Variable des Technologieobjekts: <Achsenname>.StatusBits.DriveReady)
Fehler	Der Antrieb hat nach Ausfall des Signals "Bereit" einen Fehler gemeldet. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.DriveFault)



Tabelle 9- 56 Status der Achsbewegung

Status	Beschreibung
Stillstand	Die Achse ist im Stillstand. (Variable des Technologieobjekts: <Achsenname>.StatusBits.StandStill)
Beschleunigung	Die Achse beschleunigt. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Acceleration)
Konstante Geschwindigkeit	Die Achse fährt mit konstanter Geschwindigkeit. (Variable des Technologieobjekts: <Achsenname>.StatusBits.ConstantVelocity)
Verzögerung	Die Achse verzögert (verlangsamt die Fahrt). (Variable des Technologieobjekts: <Achsenname>.StatusBits.Deceleration)

Tabelle 9- 57 Status des Bewegungsmodus

Status	Beschreibung
Positionierung	Die Achse führt eine Positionierungsaufgabe der Bewegungssteuerungsanweisung "MC_MoveAbsolute" oder "MC_MoveRelative" oder des Steuerpanels aus. (Variable des Technologieobjekts: <Achsenname>.StatusBits.PositioningCommand)
Geschwindigkeitsbefehl	Die Achse führt eine Aufgabe mit festgelegter Geschwindigkeit der Bewegungssteuerungsanweisung "MC_MoveVelocity" oder "MC_MoveJog" oder des Steuerpanels aus. (Variable des Technologieobjekts: <Achsenname>.StatusBits.SpeedCommand)
Referenzpunktfahrt	Die Achse führt eine Referenzpunktfahrt Aufgabe der Bewegungssteuerungsanweisung "MC_Home" oder des Steuerpanels aus. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Homing)

Tabelle 9- 58 Fehlerbits

Fehler	Beschreibung
Min. Softwaregrenze erreicht	Der untere Software-Endschalter wurde erreicht. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMinReached)
Min. Softwaregrenze überschritten	Der untere Software-Endschalter wurde überschritten. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMinExceeded)
Max. Softwaregrenze erreicht	Der obere Software-Endschalter wurde erreicht. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMaxReached)
Max. Softwaregrenze überschritten	Der obere Software-Endschalter wurde überschritten. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMaxExceeded)
Negative Hardwaregrenze	Der untere Hardware-Endschalter wurde angefahren. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.HwLimitMin)
Positive Hardwaregrenze	Der obere Hardware-Endschalter wurde angefahren. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.HwLimitMax)
PTO und HSC bereits verwendet	Eine zweite Achse verwendet die gleiche PTO und HSC und ist freigegeben mit "MC_Power". (Variable des Technologieobjekts: <Achsenname>.ErrorBits.HwUsed)

Fehler	Beschreibung
Konfigurationsfehler	Das Technologieobjekt "Achse" wurde falsch konfiguriert oder änderbare Konfigurationsdaten wurden während der Laufzeit des Anwenderprogramms fehlerhaft geändert. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.ConfigFault)
Allgemeiner Fehler	Ein interner Fehler ist aufgetreten. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SystemFault)

### Diagnosefunktion "Bewegungsstatus"

Mit der Diagnosefunktion "Bewegungsstatus" überwachen Sie den Bewegungsstatus der Achse. Die Diagnosefunktionsanzeige ist im Online-Modus bei aktiver Achse in der Betriebsart "Manuelle Steuerung" und "Automatiksteuerung" verfügbar.

Tabelle 9- 59 Bewegungsstatus

Status	Beschreibung
Zielposition	Das Feld "Zielposition" zeigt die aktuelle Zielposition einer aktiven Positionierungsaufgabe der Bewegungssteuerungsanweisung "MC_MoveAbsolute" oder "MC_MoveRelative" oder des Steuerpanels an. Der Wert der "Zielposition" ist nur während der Ausführung einer Positionierungsaufgabe gültig. (Variable des Technologieobjekts: <Achsenname>.MotionStatus.TargetPosition)
Aktuelle Position	Das Feld "Aktuelle Position" zeigt die aktuelle Position der Achse an. Wenn sich die Achse nicht am Referenzpunkt befindet, zeigt der Wert den Positionswert relativ zur Freigabeposition der Achse an. (Variable des Technologieobjekts: <Achsenname>.MotionStatus.Position)
Aktuelle Geschwindigkeit	Das Feld "Aktuelle Geschwindigkeit" zeigt die aktuelle Geschwindigkeit der Achse an. (Variable des Technologieobjekts: <Achsenname>.MotionStatus.Velocity)

Tabelle 9- 60 Dynamikgrenzen

Dynamikgrenze	Beschreibung
Geschwindigkeit	Das Feld "Geschwindigkeit" zeigt die konfigurierte maximale Geschwindigkeit der Achse an. (Variable des Technologieobjekts: <Achsenname>.Config.DynamicLimits.MaxVelocity)
Beschleunigung	Das Feld "Beschleunigung" zeigt die aktuell konfigurierte Beschleunigung der Achse an. (Variable des Technologieobjekts: <Achsenname>.Config.DynamicDefaults.Acceleration)
Verzögerung	Das Feld "Verzögerung" zeigt die aktuell konfigurierte Verzögerung der Achse an. (Variable des Technologieobjekts: <Achsenname>.Config.DynamicDefaults.Deceleration)

## 9.3.6 Aktive Befehle überwachen

### 9.3.6.1 Bewegungssteuerungsanweisungen mit einem Ausgangsparameter "Done" überwachen

Bewegungssteuerungsanweisungen mit dem Ausgangsparameter "Done" werden vom Eingangsparameter "Execute" gestartet und haben einen definierten Abschluss (z. B. bei der Bewegungssteuerungsanweisung "MC\_Home": Referenzpunktfahrt war erfolgreich). Die Aufgabe ist beendet und die Achse ist im Stillstand.

- Der Ausgangsparameter "Done" meldet den Wert WAHR, wenn die Aufgabe erfolgreich durchgeführt wurde.
- Die Ausgangsparameter "Busy", "CommandAborted" und "Error" melden, dass die Aufgabe noch verarbeitet wird, abgebrochen wurde oder dass ein Fehler anliegt. Die Bewegungssteuerungsanweisung "MC\_Reset" kann nicht abgebrochen werden, sie hat deshalb keinen Ausgangsparameter "CommandAborted".
  - Während der Verarbeitung der Bewegungssteuerungsaufgabe meldet der Ausgangsparameter "Busy" den Wert WAHR. Wurde die Aufgabe beendet, abgebrochen oder durch einen Fehler gestoppt, ändert sich der Wert des Ausgangsparameters "Busy" in FALSCH. Diese Änderung tritt unabhängig vom Signal am Eingangsparameter "Execute" ein.
  - Die Ausgangsparameter "Done", "CommandAborted" und "Error" melden mindestens einen Zyklus lang den Wert WAHR. Diese Statusmeldungen werden gespeichert gehalten, während der Eingangsparameter "Execute" auf WAHR ist.

Die Aufgaben der folgenden Bewegungssteuerungsanweisungen haben einen definierten Abschluss:

- MC\_Reset
- MC\_Home
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative

Das Verhalten der Statusbits wird im Folgenden für verschiedene Beispielsituationen aufgeführt.

- Das erste Beispiel zeigt das Verhalten der Achse bei einer abgeschlossenen Aufgabe. Wenn die Bewegungssteuerungsaufgabe zum Zeitpunkt des Abschlusses vollständig durchgeführt wurde, wird dies durch den Wert WAHR am Ausgangsparameter "Done" gemeldet. Der Signalzustand des Eingangsparameters "Execute" beeinflusst die Anzeigedauer im Ausgangsparameter "Done".
- Das zweite Beispiel zeigt das Verhalten der Achse bei einer abgebrochenen Aufgabe. Wenn die Bewegungssteuerungsaufgabe während der Ausführung abgebrochen wird, wird dies durch den Wert WAHR am Ausgangsparameter "CommandAborted" gemeldet. Der Signalzustand des Eingangsparameters "Execute" beeinflusst die Anzeigedauer im Ausgangsparameter "CommandAborted".
- Das dritte Beispiel zeigt das Verhalten der Achse im Fehlerfall. Tritt während der Ausführung der Bewegungssteuerungsaufgabe ein Fehler auf, wird dies durch den Wert WAHR am Ausgangsparameter "Error" gemeldet. Der Signalzustand des Eingangsparameters "Execute" beeinflusst die Anzeigedauer im Ausgangsparameter "Error".

Tabelle 9- 61 Beispiel 1 - Vollständige Durchführung der Aufgabe

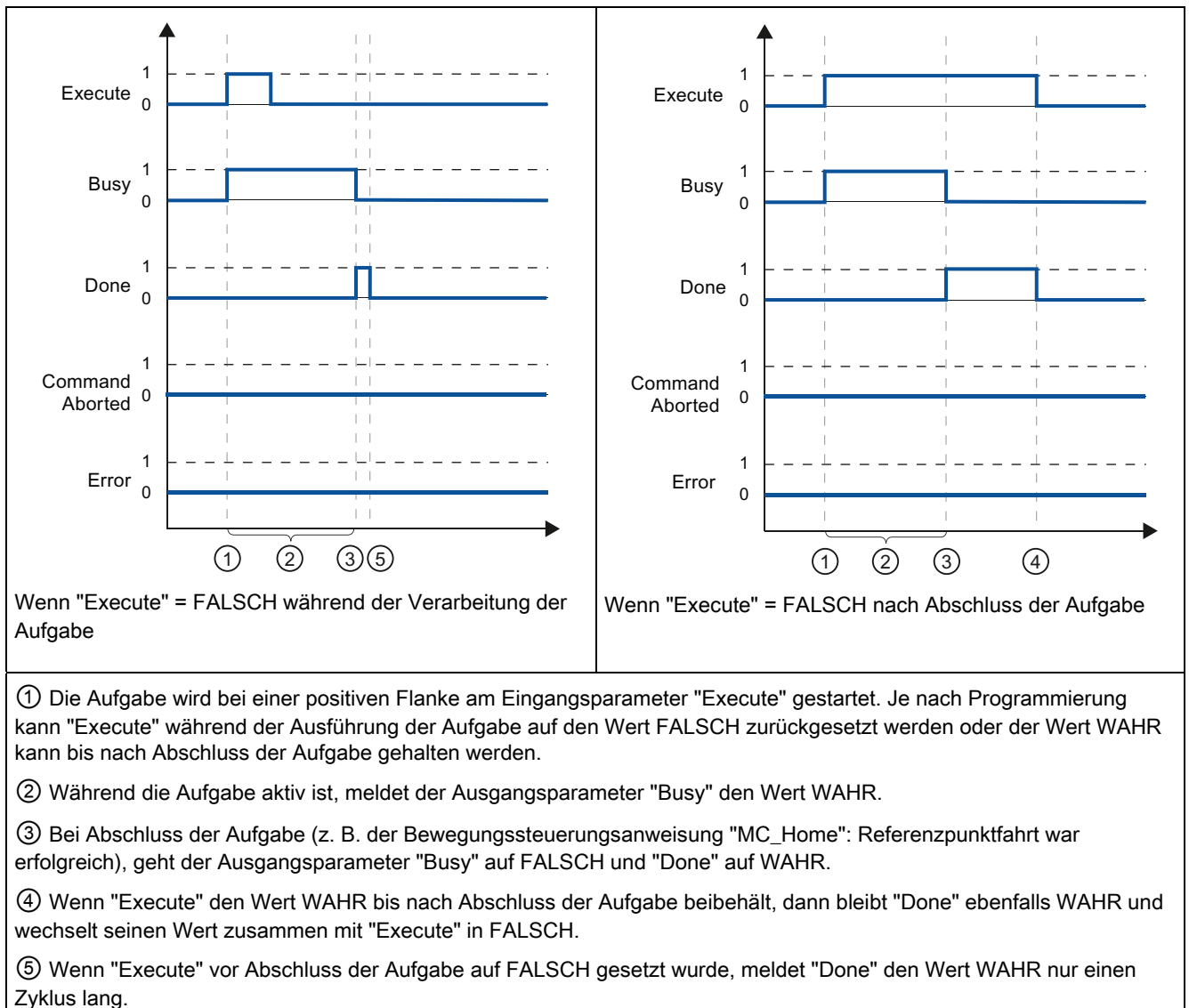
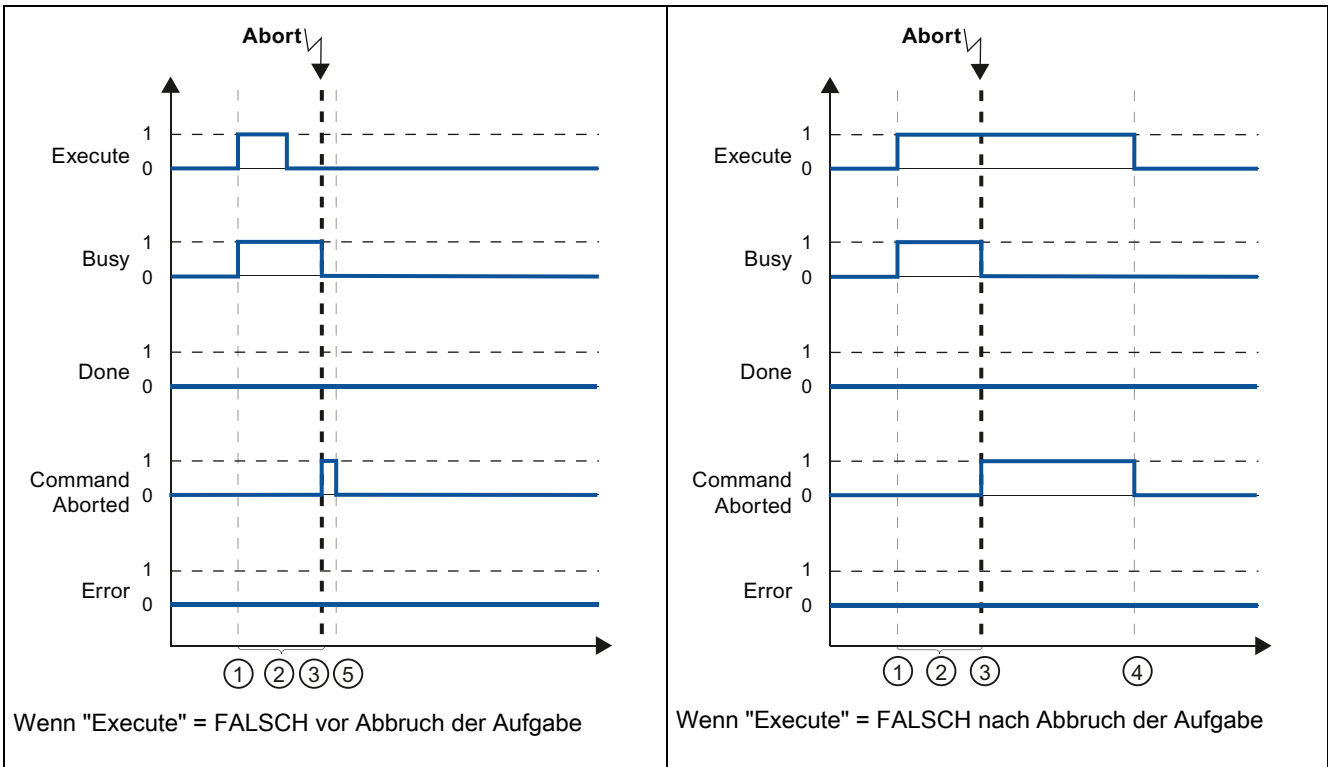
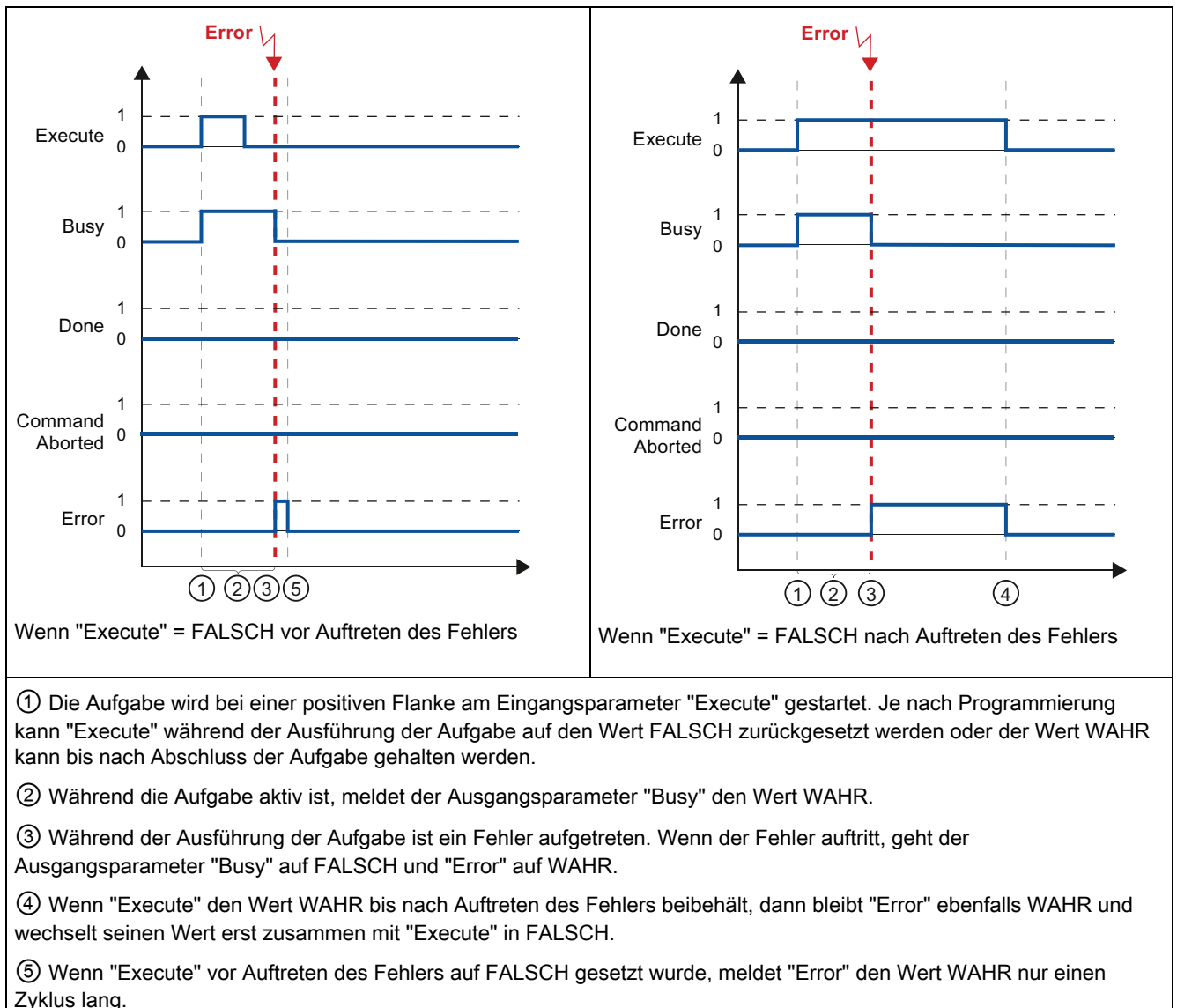


Tabelle 9- 62 Beispiel 2 - Abbruch der Aufgabe



- ① Die Aufgabe wird bei einer positiven Flanke am Eingangsparameter "Execute" gestartet. Je nach Programmierung kann "Execute" während der Ausführung der Aufgabe auf den Wert FALSCH zurückgesetzt werden oder der Wert WAHR kann bis nach Abschluss der Aufgabe gehalten werden.
- ② Während die Aufgabe aktiv ist, meldet der Ausgangsparameter "Busy" den Wert WAHR.
- ③ Die Aufgabe wird während ihrer Ausführung von einer anderen Bewegungssteuerungsaufgabe abgebrochen. Wenn die Aufgabe abgebrochen wird, geht der Ausgangsparameter "Busy" auf FALSCH und "CommandAborted" auf WAHR.
- ④ Wenn "Execute" den Wert WAHR bis nach Abbruch der Aufgabe beibehält, dann bleibt "CommandAborted" ebenfalls WAHR und wechselt seinen Wert zusammen mit "Execute" in FALSCH.
- ⑤ Wenn "Execute" vor Abbruch der Aufgabe auf FALSCH gesetzt wurde, meldet "CommandAborted" den Wert WAHR nur einen Zyklus lang.

Tabelle 9- 63 Beispiel 3 - Fehler während der Ausführung der Aufgabe



### 9.3.6.2 Anweisung MC\_Velocity überwachen

Die Aufgaben der Bewegungssteuerungsanweisung "MC\_MoveVelocity" haben konstant die vorgegebene Geschwindigkeit.

- Die Aufgaben der Bewegungssteuerungsanweisung "MC\_MoveVelocity" haben kein definiertes Ende. Das Ziel der Aufgabe ist erfüllt, wenn die parametrisierte Geschwindigkeit zum ersten Mal erreicht ist und die Achse mit konstanter Geschwindigkeit fährt. Wenn die parametrisierte Geschwindigkeit erreicht ist, wird dies durch den Wert WAHR am Ausgangsparameter "InVelocity" gemeldet.
- Die Aufgabe ist beendet, wenn die parametrisierte Geschwindigkeit erreicht ist und der Eingangsparameter "Execute" auf den Wert FALSCH gesetzt wurde. Die Achsbewegung ist jedoch mit dem Abschluss der Aufgabe noch nicht beendet. Beispiel: Die Achsbewegung kann mit der Bewegungssteuerungsaufgabe "MC\_Halt" gestoppt werden.
- Die Ausgangsparameter "Busy", "CommandAborted" und "Error" melden, dass die Aufgabe noch verarbeitet wird, abgebrochen wurde oder dass ein Fehler anliegt.
  - Während der Ausführung der Bewegungssteuerungsaufgabe meldet der Ausgangsparameter "Busy" den Wert WAHR. Wurde die Aufgabe beendet, abgebrochen oder durch einen Fehler gestoppt, ändert sich der Wert des Ausgangsparameters "Busy" in FALSCH. Diese Änderung tritt unabhängig vom Signal am Eingangsparameter "Execute" ein.
  - Die Ausgangsparameter "InVelocity", "CommandAborted" und "Error" melden mindestens einen Zyklus lang den Wert WAHR, wenn ihre Bedingungen erfüllt sind. Diese Statusmeldungen werden gespeichert gehalten, während der Eingangsparameter "Execute" auf WAHR ist.

Das Verhalten der Statusbits wird im Folgenden für verschiedene Beispielsituationen aufgeführt.

- Das erste Beispiel zeigt das Verhalten, wenn die Achse die parametrisierte Geschwindigkeit erreicht. Wenn die Bewegungssteuerungsaufgabe zu dem Zeitpunkt, zu dem die parametrisierte Geschwindigkeit erreicht wird, ausgeführt wurde, wird dies durch den Wert WAHR am Ausgangsparameter "InVelocity" gemeldet. Der Signalzustand des Eingangsparameters "Execute" beeinflusst die Anzeigedauer im Ausgangsparameter "InVelocity".
- Das zweite Beispiel zeigt das Verhalten, wenn die Aufgabe vor Erreichen der parametrisierten Geschwindigkeit abgebrochen wird. Wenn die Bewegungssteuerungsaufgabe vor Erreichen der parametrisierten Geschwindigkeit abgebrochen wird, wird dies durch den Wert WAHR am Ausgangsparameter "CommandAborted" gemeldet. Der Signalzustand des Eingangsparameters "Execute" beeinflusst die Anzeigedauer im Ausgangsparameter "CommandAborted".
- Das dritte Beispiel zeigt das Verhalten der Achse, wenn vor Erreichen der parametrisierten Geschwindigkeit ein Fehler auftritt. Tritt während der Ausführung der Bewegungssteuerungsaufgabe ein Fehler auf, bevor die parametrisierte Geschwindigkeit erreicht wurde, wird dies durch den Wert WAHR am Ausgangsparameter "Error" gemeldet. Der Signalzustand des Eingangsparameters "Execute" beeinflusst die Anzeigedauer im Ausgangsparameter "Error".



Tabelle 9- 64 Beispiel 1 - Wenn die parametrisierte Geschwindigkeit erreicht wird

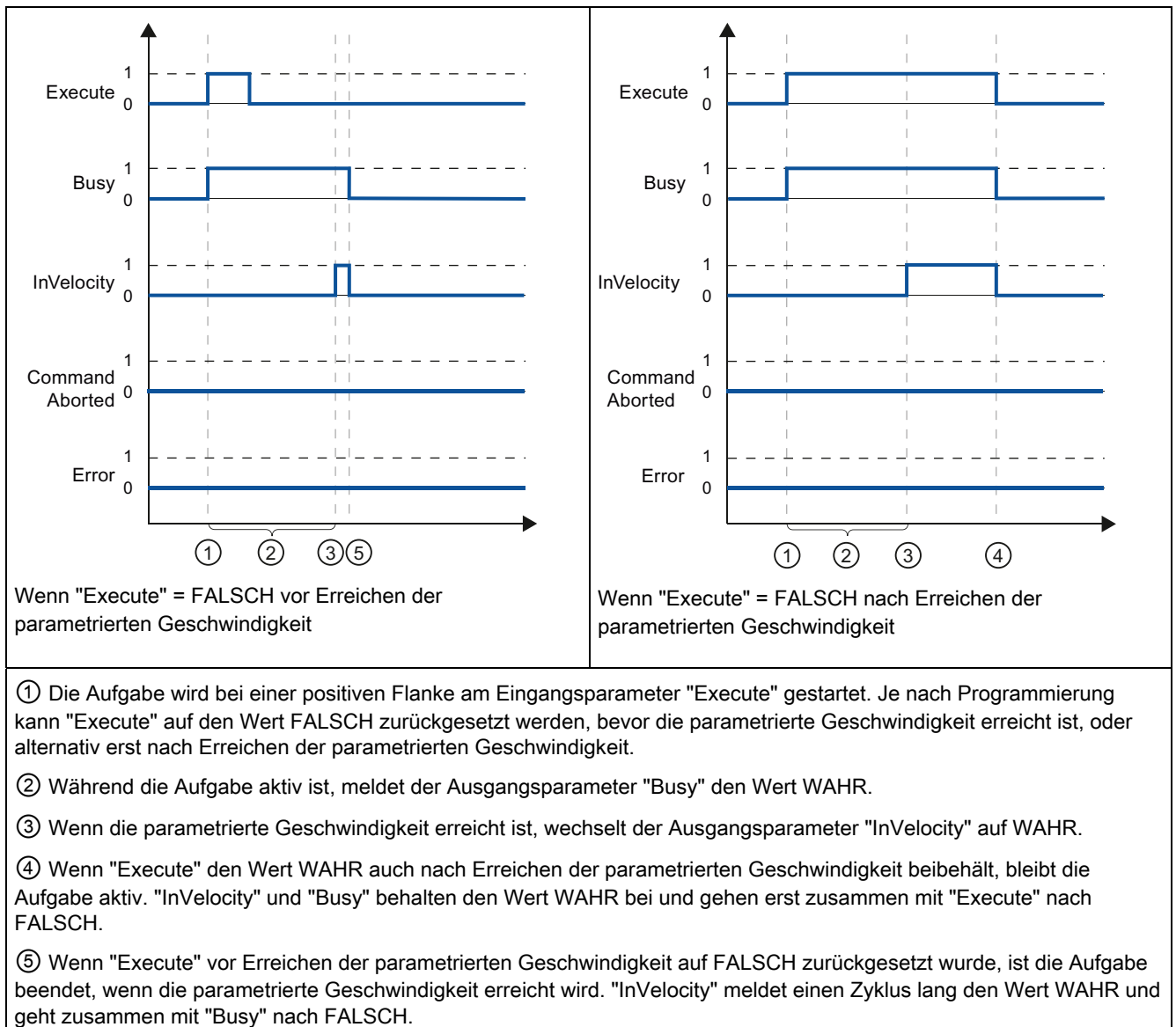
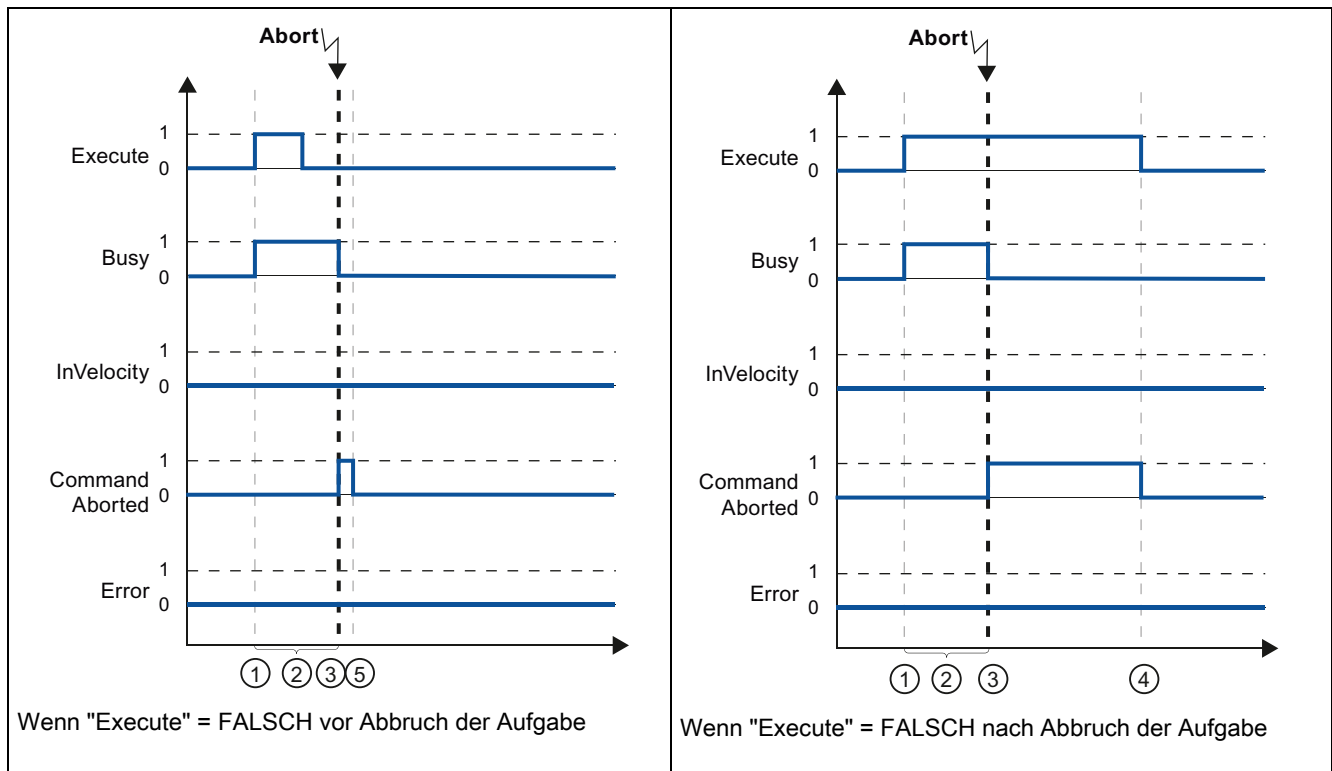


Tabelle 9- 65 Beispiel 2 - Wenn die Aufgabe vor Erreichen der parametrisierten Geschwindigkeit abgebrochen wird



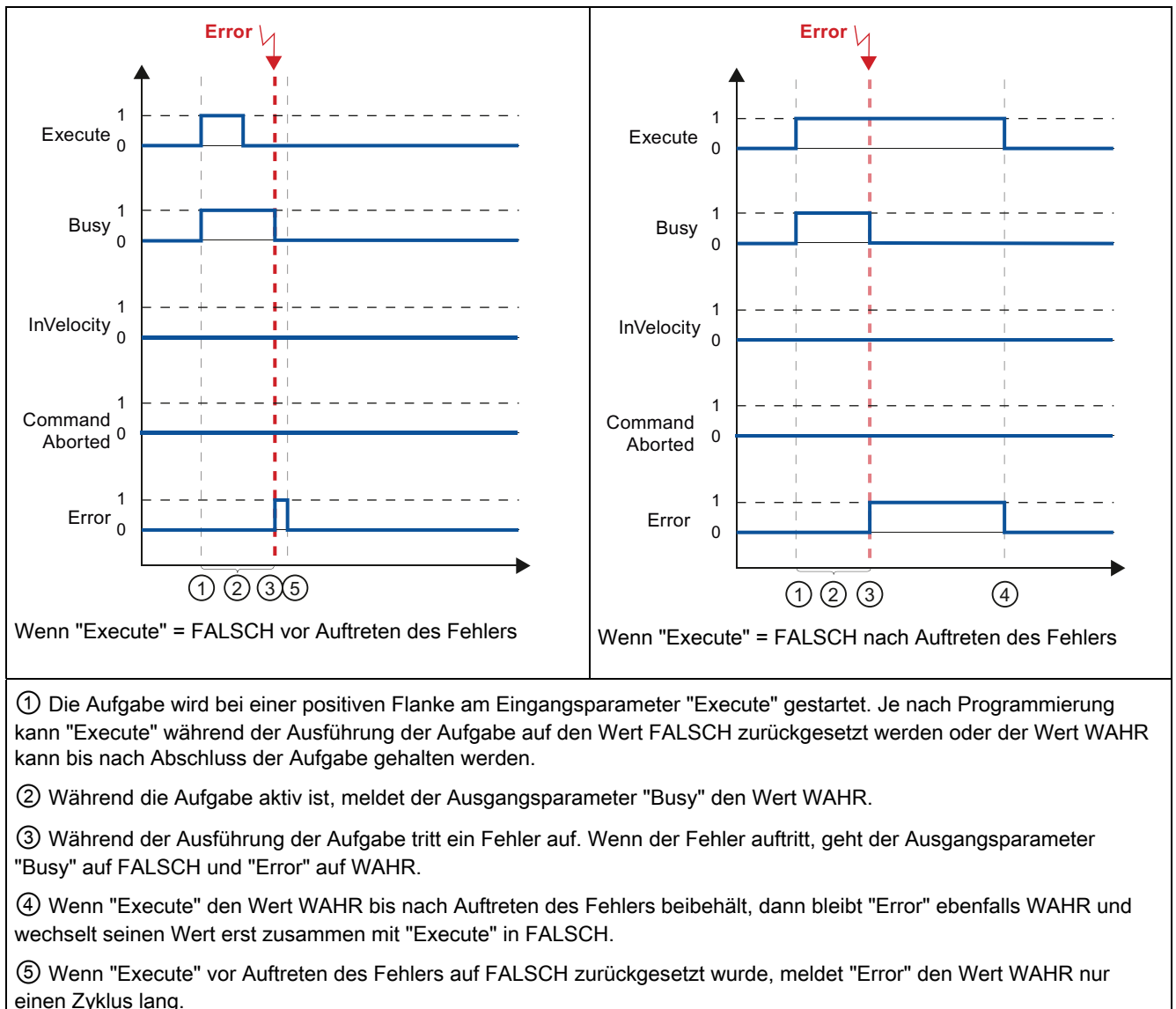
- ① Die Aufgabe wird bei einer positiven Flanke am Eingangsparameter "Execute" gestartet. Je nach Programmierung kann "Execute" während der Ausführung der Aufgabe auf den Wert FALSCH zurückgesetzt werden oder der Wert WAHR kann bis nach Abschluss der Aufgabe gehalten werden.
- ② Während die Aufgabe aktiv ist, meldet der Ausgangsparameter "Busy" den Wert WAHR.
- ③ Die Aufgabe wird während ihrer Ausführung von einer anderen Bewegungssteuerungsaufgabe abgebrochen. Wenn die Aufgabe abgebrochen wird, geht der Ausgangsparameter "Busy" auf FALSCH und "CommandAborted" auf WAHR.
- ④ Wenn "Execute" den Wert WAHR bis nach Abbruch der Aufgabe beibehält, dann bleibt "CommandAborted" ebenfalls WAHR und wechselt seinen Wert zusammen mit "Execute" in FALSCH.
- ⑤ Wenn "Execute" vor Abbruch der Aufgabe auf FALSCH zurückgesetzt wurde, meldet "CommandAborted" den Wert WAHR nur einen Zyklus lang.

**Hinweis**

Unter den folgenden Bedingungen wird ein Abbruch nicht am Ausgangsparameter "CommandAborted" angezeigt:

- Die parametrisierte Geschwindigkeit wurde erreicht, Eingangsparameter "Execute" hat den Wert FALSCH und eine neue Bewegungssteuerungsaufgabe wurde gestartet.
- Wenn die parametrisierte Geschwindigkeit erreicht ist und Eingangsparameter "Execute" den Wert FALSCH hat, ist die Aufgabe beendet. Deshalb wird der Start einer neuen Aufgabe nicht als Abbruch gemeldet.

Tabelle 9- 66 Beispiel 3 - Wenn vor Erreichen der parametrisierten Geschwindigkeit ein Fehler auftritt



### Hinweis

Unter den folgenden Bedingungen wird ein Fehler nicht am Ausgangsparameter "Error" angezeigt:

- Die parametrisierte Geschwindigkeit wurde erreicht, Eingangsparameter "Execute" hat den Wert FALSCH und ein Achsenfehler tritt auf (z. B. Software-Endschalter wird angefahren).
- Wenn die parametrisierte Geschwindigkeit erreicht ist und Eingangsparameter "Execute" den Wert FALSCH hat, ist die Aufgabe beendet. Nach Abschluss der Aufgabe wird der Achsenfehler nur in der Bewegungssteuerungsanweisung "MC\_Power" gemeldet.

### 9.3.6.3 Anweisung MC\_MoveJog überwachen

Die Aufgaben der Bewegungssteuerungsanweisung "MC\_MoveJog" implementieren einen Tipbetrieb.

- Die Bewegungssteuerungsaufgaben "MC\_MoveJog" haben kein definiertes Ende. Das Ziel der Aufgabe ist erfüllt, wenn die parametrisierte Geschwindigkeit zum ersten Mal erreicht ist und die Achse mit konstanter Geschwindigkeit fährt. Wenn die parametrisierte Geschwindigkeit erreicht ist, wird dies durch den Wert WAHR am Ausgangsparameter "InVelocity" gemeldet.
- Der Auftrag ist beendet, wenn der Eingangsparameter "JogForward" oder "JogBackward" auf den Wert FALSCH gesetzt wurde und die Achse zum Stillstand gekommen ist.
- Die Ausgangsparameter "Busy", "CommandAborted" und "Error" melden, dass die Aufgabe noch verarbeitet wird, abgebrochen wurde oder dass ein Fehler anliegt.
  - Während der Verarbeitung der Bewegungssteuerungsaufgabe meldet der Ausgangsparameter "Busy" den Wert WAHR. Wurde die Aufgabe beendet, abgebrochen oder durch einen Fehler gestoppt, ändert sich der Wert des Ausgangsparameters "Busy" in FALSCH.
  - Der Ausgangsparameter "InVelocity" meldet den Zustand WAHR, solange sich die Achse mit der parametrisierten Geschwindigkeit bewegt. Die Ausgangsparameter "CommandAborted" und "Error" melden mindestens einen Zyklus lang den Status. Diese Statusmeldungen werden gespeichert gehalten, solange einer der Eingangsparameter "JogForward" oder "JogBackward" auf WAHR ist.

Das Verhalten der Statusbits wird im Folgenden für verschiedene Beispielsituationen aufgeführt.

- Das erste Beispiel zeigt das Verhalten der Achse, wenn die parametrisierte Geschwindigkeit erreicht ist und gehalten wird. Wenn die Bewegungssteuerungsaufgabe zu dem Zeitpunkt, zu dem die parametrisierte Geschwindigkeit erreicht wird, ausgeführt wurde, wird dies durch den Wert WAHR am Ausgangsparameter "InVelocity" gemeldet.
- Das zweite Beispiel zeigt das Verhalten der Achse, wenn die Aufgabe abgebrochen wird. Wenn die Bewegungssteuerungsaufgabe während der Ausführung abgebrochen wird, wird dies durch den Wert WAHR am Ausgangsparameter "CommandAborted" gemeldet. Das Verhalten ist unabhängig davon, ob die parametrisierte Geschwindigkeit erreicht wurde oder nicht.
- Das dritte Beispiel zeigt das Verhalten der Achse im Fehlerfall. Tritt während der Ausführung der Bewegungssteuerungsaufgabe ein Fehler auf, wird dies durch den Wert WAHR am Ausgangsparameter "Error" gemeldet. Das Verhalten ist unabhängig davon, ob die parametrisierte Geschwindigkeit erreicht wurde oder nicht.

Tabelle 9- 67 Beispiel 1 - Wenn die parametrisierte Geschwindigkeit erreicht und gehalten wird

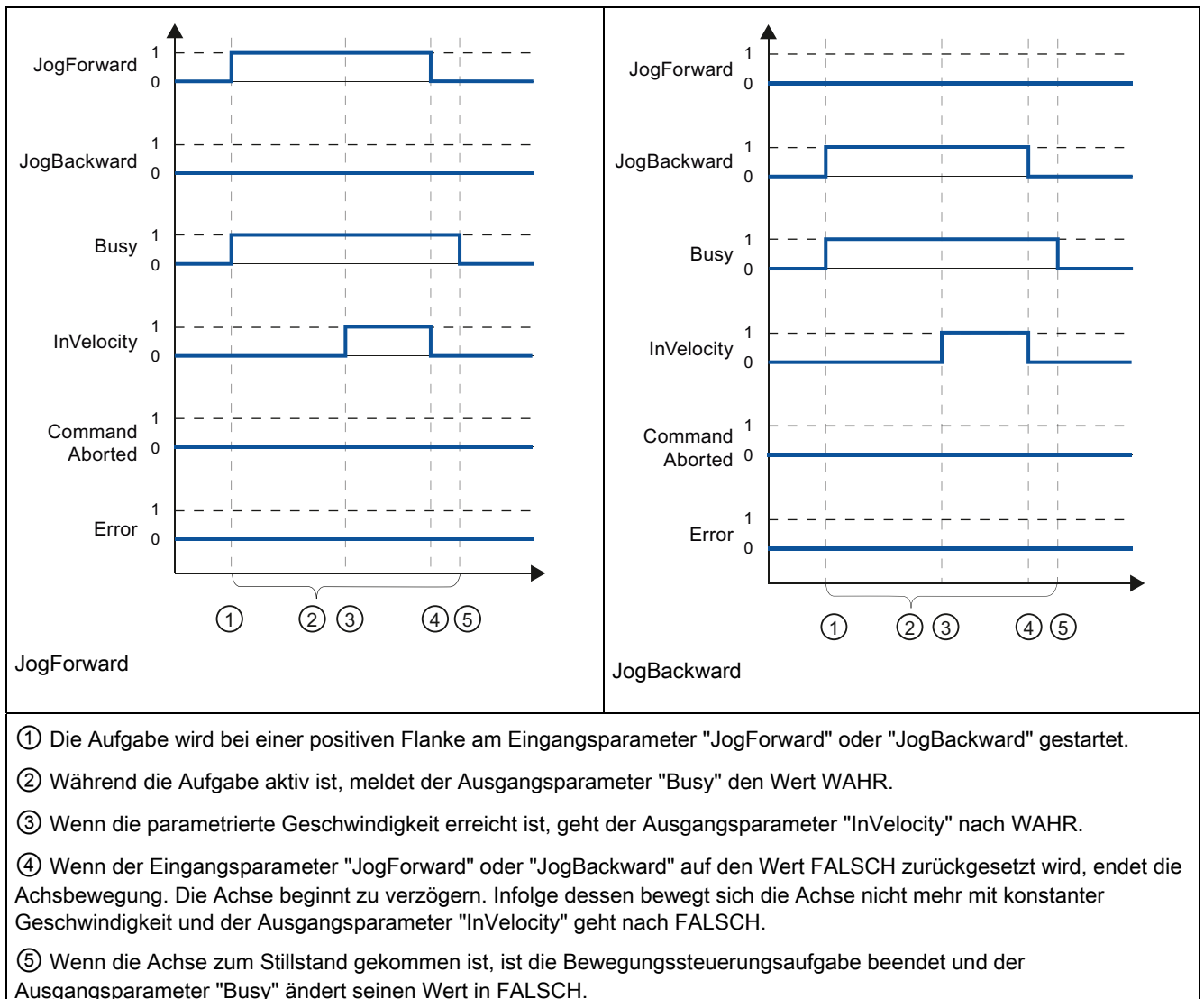
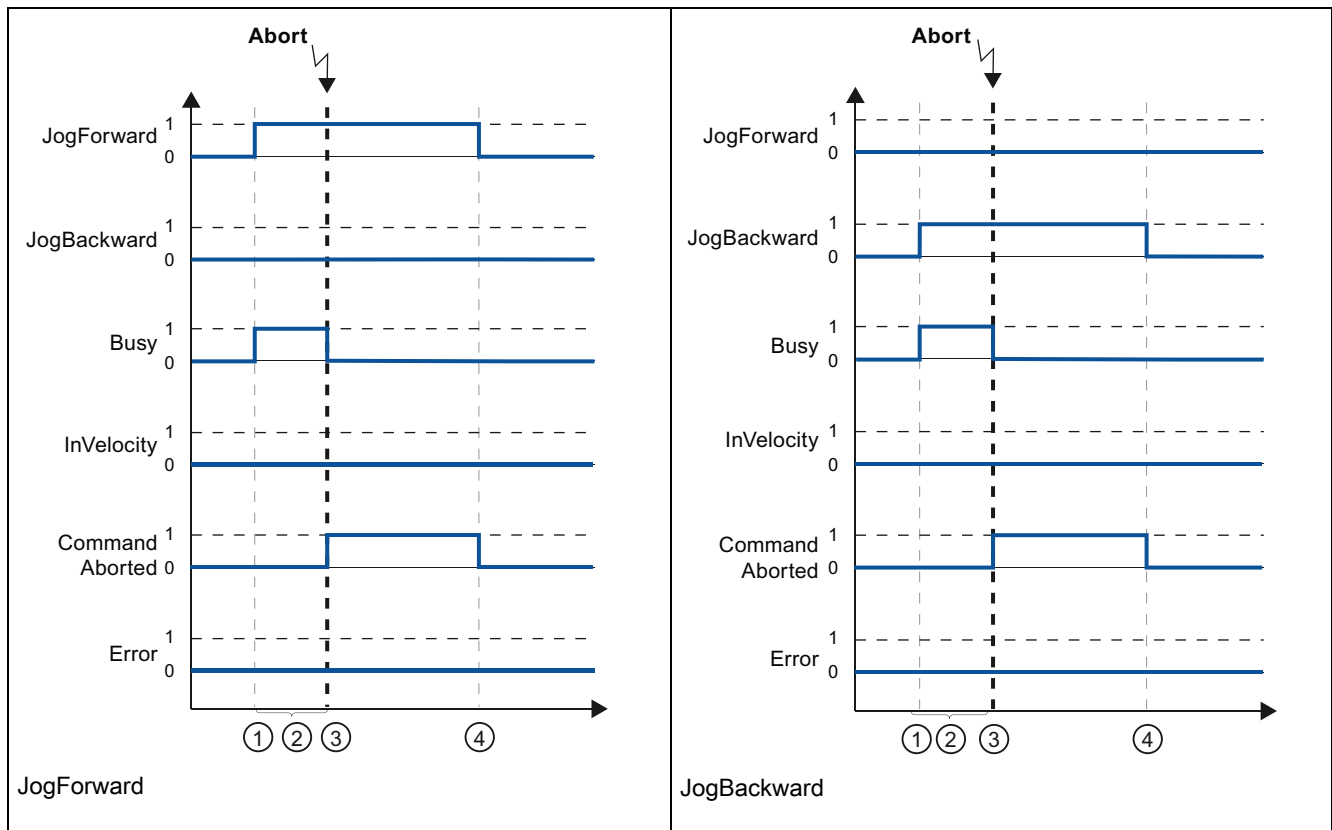


Tabelle 9- 68 Beispiel 2 - Wenn die Aufgabe während der Ausführung abgebrochen wird



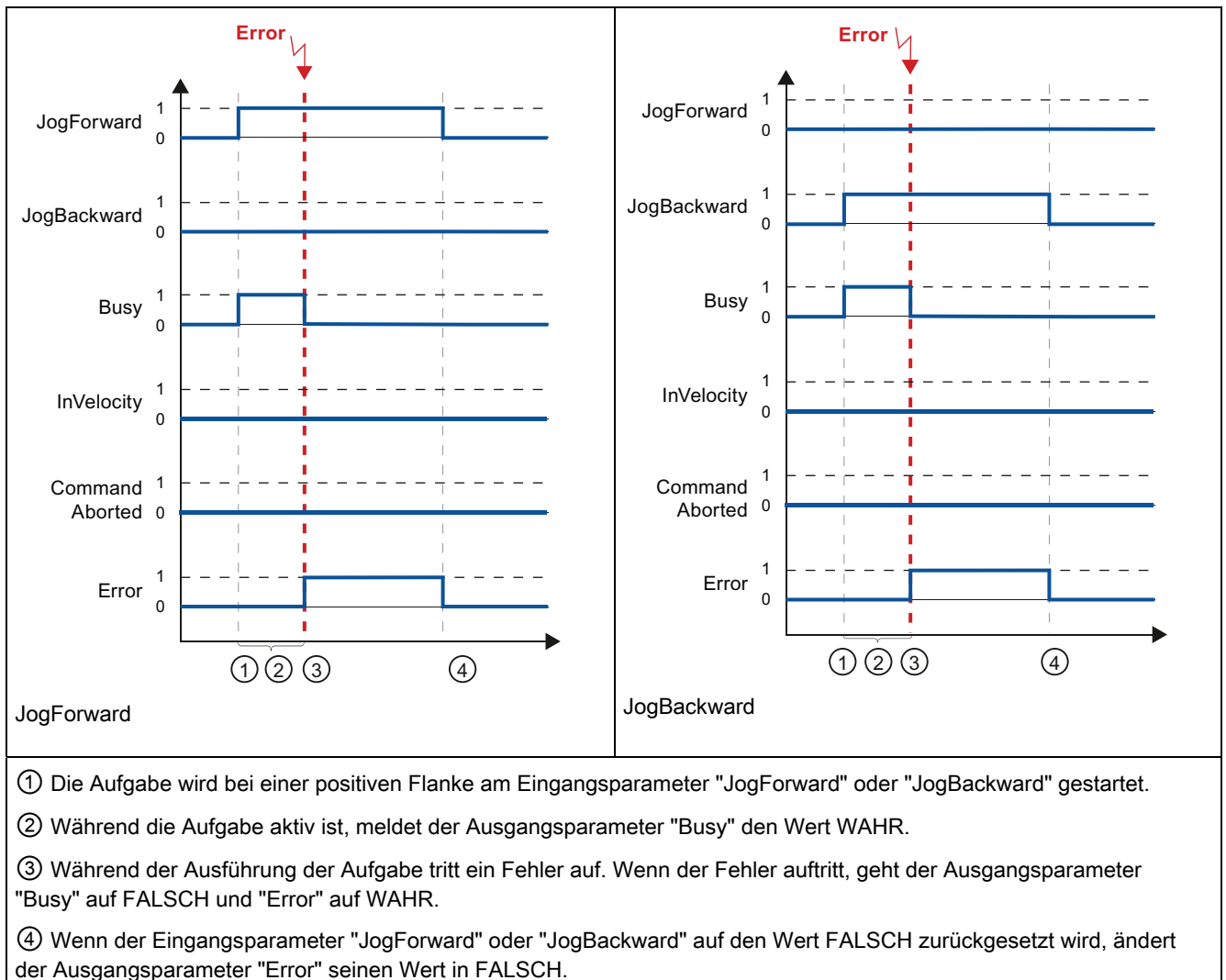
- ① Die Aufgabe wird bei einer positiven Flanke am Eingangsparameter "JogForward" oder "JogBackward" gestartet.
- ② Während die Aufgabe aktiv ist, meldet der Ausgangsparameter "Busy" den Wert WAHR.
- ③ Die Aufgabe wird während ihrer Ausführung von einer anderen Bewegungssteuerungsaufgabe abgebrochen. Wenn die Aufgabe abgebrochen wird, geht der Ausgangsparameter "Busy" auf FALSCH und "CommandAborted" auf WAHR.
- ④ Wenn der Eingangsparameter "JogForward" oder "JogBackward" auf den Wert FALSCH zurückgesetzt wird, ändert der Ausgangsparameter "CommandAborted" seinen Wert in FALSCH.

**Hinweis**

Der Abbruch der Aufgabe wird am Ausgangsparameter "CommandAborted" nur einen Ausführungszyklus lang gemeldet, wenn alle folgenden Bedingungen erfüllt sind:

Die Eingangsparameter "JogForward" und "JogBackward" haben den Wert FALSCH (doch die Achse verzögert noch) und eine neue Bewegungssteuerungsaufgabe wird gestartet.

Tabelle 9- 69 Beispiel 3 - Wenn während der Ausführung der Aufgabe ein Fehler aufgetreten ist



- ① Die Aufgabe wird bei einer positiven Flanke am Eingangsparameter "JogForward" oder "JogBackward" gestartet.
- ② Während die Aufgabe aktiv ist, meldet der Ausgangsparameter "Busy" den Wert WAHR.
- ③ Während der Ausführung der Aufgabe tritt ein Fehler auf. Wenn der Fehler auftritt, geht der Ausgangsparameter "Busy" auf FALSCH und "Error" auf WAHR.
- ④ Wenn der Eingangsparameter "JogForward" oder "JogBackward" auf den Wert FALSCH zurückgesetzt wird, ändert der Ausgangsparameter "Error" seinen Wert in FALSCH.

### Hinweis

Der aufgetretene Fehler wird am Ausgangsparameter "Error" nur einen Ausführungszyklus lang gemeldet, wenn alle folgenden Bedingungen erfüllt sind:

Die Eingangsparameter "JogForward" und "JogBackward" haben den Wert FALSCH (doch die Achse verzögert noch) und ein neuer Fehler tritt auf (z. B., Software-Endschalter wird angefahren).





## Kommunikation

Die S7-1200 bietet mehrere Arten der Kommunikation zwischen CPUs und Programmiergeräten, HMI-Geräten und anderen CPUs.

### PROFINET

PROFINET wird für den Austausch von Daten über das Anwenderprogramm mit anderen Kommunikationspartnern über Ethernet verwendet:

- Die CPU bietet folgende PROFINET- und PROFIBUS-Unterstützung:
  - In V3.0 unterstützt PROFINET 16 IO-Geräte mit maximal 256 Submodulen. PROFIBUS gestattet 3 unabhängige PROFIBUS DP-Master und unterstützt 32 IO-Geräte mit maximal 512 Submodulen je IO-Gerät.
  - In V2.2 unterstützt PROFINET 8 IO-Geräte mit maximal 128 Submodulen (wenn acht oder weniger PROFIBUS-Slaves oder -Submodule konfiguriert sind). PROFIBUS unterstützt maximal 16 PROFIBUS IO-Geräte an einem Master mit maximal 256 Submodulen je IO-Gerät.
- S7-Kommunikation
- User Datagram Protocol (UDP)
- ISO on TCP (RFC 1006)
- Transport Control Protocol (TCP)

### PROFINET RT IO-Steuerung

Als E/A-Steuerung über PROFINET RT bietet die CPU die folgende Unterstützung im lokalen PN-Netzwerk oder über einen PN/PN-Koppler (Verbund). Ausführliche Informationen hierzu finden Sie unter PROFIBUS and PROFINET International, PI ([www.profinet.com](http://www.profinet.com)):

- In V3.0 kommuniziert die S7-1200 mit bis zu 16 PN-Geräten.
- In V2.2 kommuniziert die S7-1200 mit bis zu 8 PN-Geräten.

### PROFIBUS

PROFIBUS wird für den Austausch von Daten über das Anwenderprogramm mit anderen Kommunikationspartnern über das PROFIBUS-Netzwerk verwendet:

- Beim CM 1242-5 fungiert die CPU als PROFIBUS DP-Slave.
- Beim CM 1243-5 fungiert die CPU als PROFIBUS DP-Master der Klasse 1.
- In V3.0 sind PROFIBUS DP-Slaves, PROFIBUS DP-Master und ASi (die 3 Kommunikationsmodule auf der linken Seite) und PROFINET voneinander getrennt.

- In V2.2. bietet die CPU folgende PROFINET- und PROFIBUS-Unterstützung:
  - Insgesamt 16 Geräte und 256 Submodule mit maximal 16 PROFIBUS DP-Slavegeräten und 256 Submodulen (falls keine PROFINET IO-Geräte oder -Submodule konfiguriert sind).

---

#### Hinweis

In V2.2 sind in der Summe von 16 PROFINET- und PROFIBUS-Geräten folgende Geräte enthalten:

- Die an den PROFIBUS DP-Master (CM 1243-5) angeschlossenen PROFIBUS DP-Slavemodule
- Alle an die CPU angeschlossenen PROFIBUS DP-Slavemodule (CM 1242-5)
- Alle über den PROFINET-Port an die CPU angeschlossenen PROFINET-Geräte

So ist beispielsweise in einer Konfiguration mit drei PROFIBUS-CMs (mit einem CM 1243-5-Master und zwei CM 1242-5-Slavemodulen) die maximale Anzahl von Slavemodulen, auf die der PROFIBUS DP-Master (CM 1243-5) zugreifen kann, auf 14 verringert.

---

- AS-i: Der S7-1200 CM 1243-2 AS-i Master ermöglicht die Anbindung eines AS-i-Netzwerks an eine S7-1200 CPU.
- CPU-zu-CPU-Kommunikation über S7

### Teleservice-Kommunikation

Bei TeleService über GPRS kommuniziert eine Engineering-Station, auf der STEP 7 installiert ist, über das GSM-Netz und das Internet mit einer SIMATIC S7-1200 Station mit einem CP 1242-7. Die Verbindung läuft über einen Telecontrol-Server, der als Vermittler dient und mit dem Internet verbunden ist.

## 10.1 Anzahl der unterstützten asynchronen Kommunikationsverbindungen

Die CPU unterstützt die folgende maximale Anzahl von gleichzeitigen, asynchronen Kommunikationsverbindungen für PROFINET und PROFIBUS:

- 8 Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND\_C, TRCV\_C, TCON, TDISCON, TSEND und TRCV.
- 3 CPU-zu-CPU-S7-Verbindungen für Server-GET/PUT-Daten
- 8 CPU-zu-CPU-S7-Verbindungen für Client-GET/PUT-Daten

---

### Hinweis

S7-1200, S7-300 und S7-400 CPUs nutzen die Anweisungen GET und PUT für die S7-Kommunikation von CPU zu CPU. Eine S7-200 CPU verwendet ETHx\_XFER-Anweisungen für die S7-Kommunikation von CPU zu CPU.

---

- HMI-Verbindungen: Die CPU stellt zweckbestimmte HMI-Verbindungen bereit, um bis zu 3 HMI-Geräte zu unterstützen. (Sie können bis zu 2 SIMATIC Comfort Panels haben.) Wie viele HMI-Geräte insgesamt unterstützt werden, hängt von den Typen der HMI-Panels in Ihrer Konfiguration ab. Sie können beispielsweise bis zu drei SIMATIC Basic Panels an Ihre CPU anschließen, oder Sie können bis zu zwei SIMATIC Comfort Panels und ein zusätzliches Basic Panel anschließen.
- PG-Verbindungen: Die CPU bietet Verbindungen, um 1 Programmiergerät (PG) zu unterstützen.
- Webserver-Verbindungen (HTTP): Die CPU bietet Verbindungen für den Webserver.

## 10.2 PROFINET

### 10.2.1 Verbindung zwischen lokaler und Partner-CPU

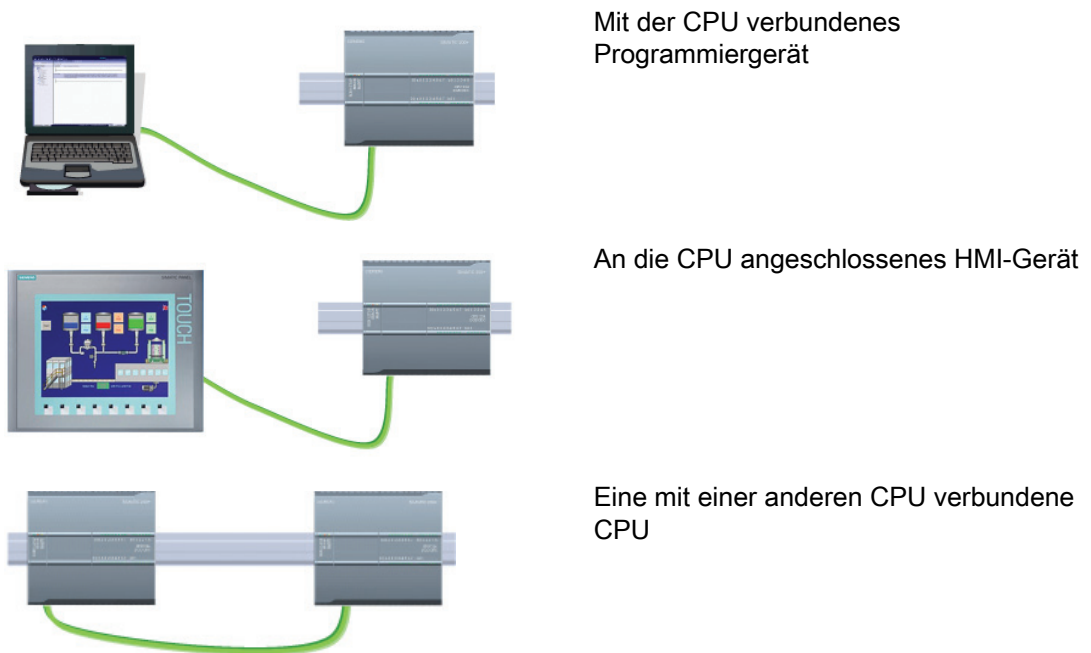
Eine Verbindung zwischen lokaler CPU und Partner-CPU (entfernt) definiert eine logische Zuweisung von zwei Kommunikationspartnern für die Herstellung von Kommunikationsdiensten. Eine Verbindung definiert folgende Elemente:

- Beteiligte Kommunikationspartner (einer aktiv, einer passiv)
- Die Art der Verbindung (z. B. eine PLC-, HMI- oder Geräteverbindung)
- Den Verbindungspfad

Die Kommunikationspartner führen die Anweisungen zum Einrichten und Herstellen der Kommunikationsverbindung aus. Mit Parametern geben Sie die aktiven und passiven Endpunkte der Kommunikation an. Nach dem Einrichten und Aufbauen der Verbindung wird sie automatisch von der CPU gehalten und überwacht. Weitere Informationen zum Konfigurieren der Parameter für die Verbindung finden Sie im Abschnitt "Verbindung zwischen lokaler und Partner-CPU konfigurieren" (Seite 138).

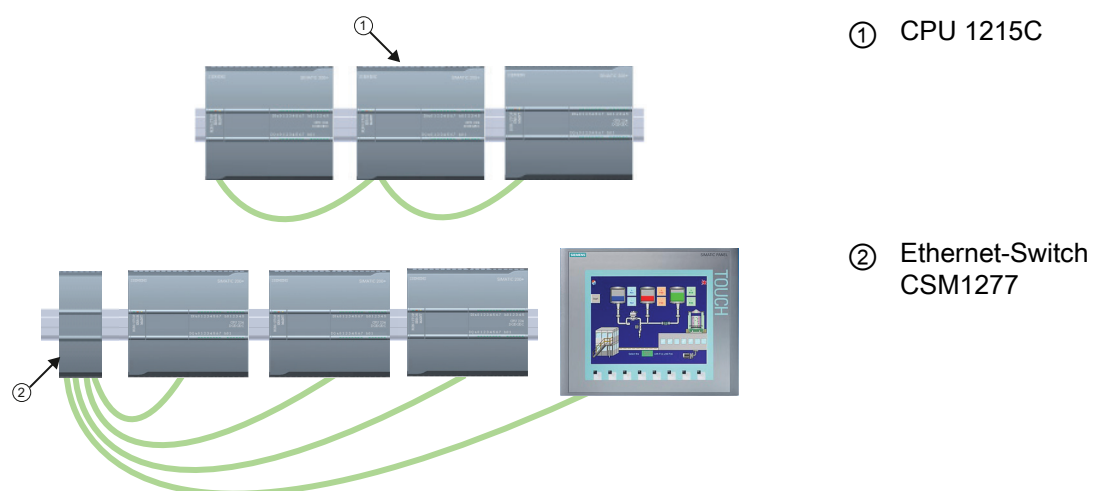
Wird die Verbindung beendet (z. B. durch eine Leitungsstörung), so versucht der aktive Partner, sie wieder aufzubauen. Sie müssen die Kommunikationsanweisung nicht erneut ausführen.

Die CPU kann mit anderen CPUs, mit Programmiergeräten, mit HMI-Geräten und Siemens-fremden Geräten über herkömmliche TCP-Kommunikationsprotokolle kommunizieren.



## Ethernet-Switch

Der PROFINET-Anschluss an der CPU 1211C, 1212C und 1214C enthält keinen Ethernet-Switch. Eine direkte Verbindung zwischen einem Programmiergerät oder HMI-Gerät und einer CPU erfordert keinen Ethernet-Switch. Ein Netzwerk mit mehr als zwei CPUs oder HMI-Geräten jedoch benötigt einen Ethernet-Switch.



Die CPU 1215C hat einen integrierten Ethernet-Switch mit zwei Ports. Sie können ein Netzwerk mit einer CPU 1215C und zwei anderen S7-1200 CPUs aufbauen. Zum Anschließen mehrerer CPUs und HMI-Geräte können Sie auch einen auf dem Baugruppenträger montierten Ethernet-Switch CSM1277 mit 4 Ports verwenden.

## 10.2.2 Offene Benutzerkommunikation

### 10.2.2.1 Verbindungs-IDs für die PROFINET-Anweisungen

Wenn Sie die PROFINET-Anweisung TSEND\_C, TRCV\_C oder TCON in Ihr Anwenderprogramm einfügen, erstellt STEP 7 einen Instanz-DB für die Konfiguration des Kommunikationskanals (oder der Verbindung) zwischen den Geräten. Die Parameter der Verbindung konfigurieren Sie in den "Eigenschaften" der Anweisung. Unter den Parametern ist auch die Verbindungs-ID der Verbindung.

- Die Verbindungs-ID muss für die CPU eindeutig sein. Jede Verbindung, die Sie erstellen, benötigt einen anderen DB und eine andere Verbindungs-ID.
- Die lokale CPU und die Partner-CPU können dieselbe Verbindungs-ID für dieselbe Verbindung verwenden, doch die Verbindungs-IDs müssen nicht übereinstimmen. Die Verbindungs-ID ist nur für die PROFINET-Anweisungen innerhalb des Anwenderprogramms der jeweiligen CPU relevant.
- Für die Verbindungs-ID der CPU können Sie jede beliebige Nummer verwenden. Wenn Sie die Verbindungs-IDs jedoch sequenziell von "1" an vergeben, können Sie leichter die Anzahl der verwendeten Verbindungen für eine bestimmte CPU nachvollziehen.

---

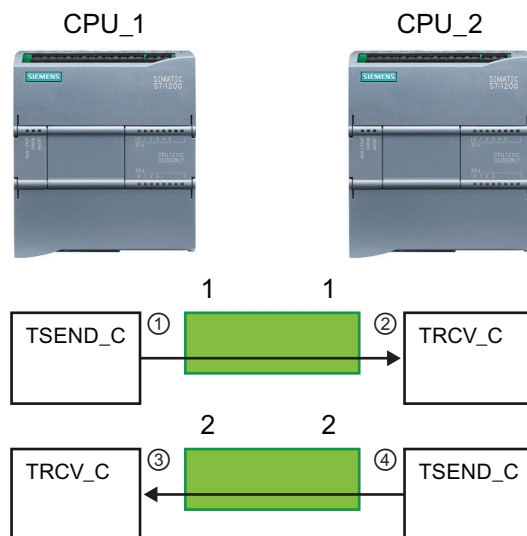
#### Hinweis

Jede Anweisung TSEND\_C, TRCV\_C oder TCON in Ihrem Anwenderprogramm erstellt eine neue Verbindung. Es ist wichtig, dass Sie für jede Verbindung die korrekte Verbindungs-ID verwenden.

---

Das folgende Beispiel zeigt die Kommunikation zwischen zwei CPUs, die zwei getrennte Verbindungen zum Senden und Empfangen von Daten nutzen.

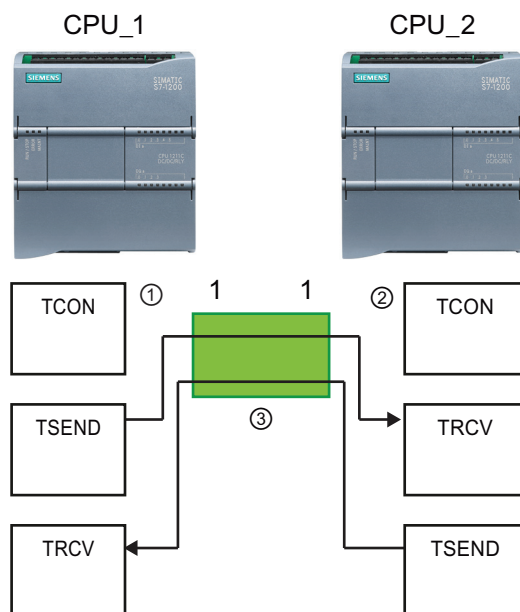
- Die Anweisung TSEND\_C in CPU\_1 bezieht sich auf die Anweisung TRCV\_V in CPU\_2 über die erste Verbindung ("Verbindungs-ID 1" bei beiden CPUs, CPU\_1 und CPU\_2).
- Die Anweisung TRCV\_C in CPU\_1 bezieht sich auf die Anweisung TSEND\_C in CPU\_2 über die zweite Verbindung ("Verbindungs-ID 2" bei beiden CPUs, CPU\_1 und CPU\_2).



- ① TSEND\_C in CPU\_1 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID zu (Verbindungs-ID 1 bei CPU\_1).
- ② TRCV\_C in CPU\_2 erstellt die Verbindung für CPU\_2 und weist die Verbindungs-ID zu (Verbindungs-ID 1 bei CPU\_2).
- ③ TRCV\_C in CPU\_1 erstellt eine zweite Verbindung für CPU\_1 und weist dieser Verbindung eine andere Verbindungs-ID zu (Verbindungs-ID 2 bei CPU\_1).
- ④ TSEND\_C in CPU\_2 erstellt eine zweite Verbindung und weist dieser Verbindung eine andere Verbindungs-ID zu (Verbindungs-ID 2 bei CPU\_2).

Das folgende Beispiel zeigt die Kommunikation zwischen zwei CPUs, die nur eine Verbindung zum Senden und Empfangen von Daten nutzen.

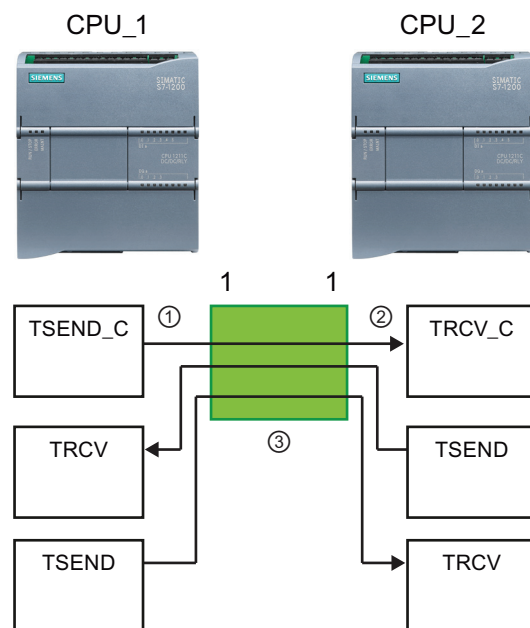
- Jede CPU nutzt eine Anweisung TCON, um die Verbindung zwischen den beiden CPUs zu konfigurieren.
- Die Anweisung TSEND in CPU\_1 bezieht sich auf die Anweisung TRCV in CPU\_2 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU\_1 konfiguriert wurde. Die Anweisung TRCV in CPU\_2 bezieht sich auf die Anweisung TSEND in CPU\_1 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU\_2 konfiguriert wurde.
- Die Anweisung TSEND in CPU\_2 bezieht sich auf die Anweisung TRCV in CPU\_1 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU\_2 konfiguriert wurde. Die Anweisung TRCV in CPU\_1 bezieht sich auf die Anweisung TSEND in CPU\_2 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU\_1 konfiguriert wurde.



- ① TCON in CPU\_1 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID in CPU\_1 zu (ID = 1).
- ② TCON in CPU\_2 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID in CPU\_2 zu (ID = 1).
- ③ TSEND und TRCV in CPU\_1 nutzen die von TCON in CPU\_1 erstellte Verbindungs-ID (ID = 1). TSEND und TRCV in CPU\_2 nutzen die von TCON in CPU\_2 erstellte Verbindungs-ID (ID = 1).



Wie das folgende Beispiel zeigt, können Sie auch mit einzelnen Anweisungen TSEND und TRCV über eine von einer Anweisung TSEND\_C oder TRCV\_C erstellte Verbindung kommunizieren. Die Anweisungen TSEND und TRCV erstellen selbst keine neue Verbindung, deshalb müssen sie den DB und die Verbindungs-ID nutzen, die von einer Anweisung TSEND\_C, TRCV\_C oder TCON erstellt wurden.



- ① TSEND\_C in CPU\_1 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID zu (ID = 1).
- ② TRCV\_C in CPU\_2 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID in CPU\_2 zu (ID = 1).
- ③ TSEND und TRCV in CPU\_1 nutzen die von TSEND\_C in CPU\_1 erstellte Verbindungs-ID (ID = 1). TSEND und TRCV in CPU\_2 nutzen die von TRCV\_C in CPU\_2 erstellte Verbindungs-ID (ID = 1).

## Siehe auch

Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 138)

### 10.2.2.2 Protokolle

Der integrierte PROFINET-Port der CPU unterstützt mehrere Kommunikationsstandards über ein Ethernet-Netzwerk:

- Transport Control Protocol (TCP)
- ISO on TCP (RFC 1006)
- User Datagram Protocol (UDP)

Tabelle 10- 1 Protokolle und Kommunikationsanweisungen

Protokoll	Verwendungsbeispiele	Eintragen der Daten in den Empfangsbereich	Kommunikationsanweisungen	Adressierungsart
TCP	CPU-zu-CPU-Kommunikation Transport von Telegrammen	Ad-hoc-Modus	Nur TRCV_C und TRCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) Portnummern zu
		Datenempfang mit angegebener Länge	TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV	
ISO on TCP	CPU-zu-CPU-Kommunikation Fragmentierung und Zusammensetzung von Meldungen	Ad-hoc-Modus	Nur TRCV_C und TRCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) TSAPs zu
		Protokollgesteuert	TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV	
UDP	CPU-zu-CPU-Kommunikation Kommunikation im Anwenderprogramm	User Datagram Protocol	TUSEND und TURCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) Portnummern zu, es handelt sich jedoch nicht um eine dedizierte Verbindung
S7-Kommunikation	CPU-zu-CPU-Kommunikation Daten aus einer CPU lesen/in eine CPU schreiben	Datenübertragung und -empfang mit angegebener Länge	GET und PUT	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) TSAPs zu
PROFINET RT	Kommunikation zwischen CPU und PROFINET IO-Gerät	Datenübertragung und -empfang mit angegebener Länge	Integriert	Integriert

### 10.2.2.3 Ad-hoc-Modus

Typischerweise empfangen die Protokolle TCP und ISO-on-TCP Datenpakete mit fest angegebener Länge von 1 bis 8192 Byte. Die Kommunikationsanweisungen TRCV\_C und TRCV jedoch bieten auch einen Ad-hoc-Kommunikationsmodus, in dem Datenpakete variabler Länge von 1 bis 1472 Byte empfangen werden können.

---

#### Hinweis

Wenn Sie die Daten in einem "optimierten" DB (nur symbolisch adressierbar) speichern, können Sie Daten nur in Arrays der Datentypen Byte, Char, USInt und SInt empfangen.

---

Um die Anweisung TRCV\_C oder TRCV für den Ad-hoc-Modus zu konfigurieren, setzen Sie den Parameter LEN auf 65535 (0xFFFF).

Wenn Sie die Anweisung TRCV\_C oder TRCV nicht sehr häufig im Ad-hoc-Modus aufrufen, können Sie in einem Aufruf mehrere Pakete empfangen. Beispiel: Wenn Sie fünf 100-Byte-Pakete in einem Aufruf empfangen möchten, liefert TCP diese fünf Pakete als ein 500-Byte-Paket, dagegen unterteilt ISO-on-TCP die Pakete in fünf 100-Byte-Pakete.

#### 10.2.2.4 TCP und ISO on TCP

Transport Control Protocol (TCP) ist ein Standardprotokoll, beschrieben von RFC 793: Transmission Control Protocol. Wesentlicher Zweck des TCP ist die Bereitstellung einer zuverlässigen, sicheren Verbindung zwischen zwei Paaren von Prozessen. Dieses Protokoll hat die folgenden Merkmale:

- Effizientes Kommunikationsprotokoll, da es eng mit der Hardware verknüpft ist
- Geeignet für mittelgroße bis große Datenmengen (bis 8192 Byte)
- Bietet deutlich mehr Leistungsmerkmale für Anwendungen, insbesondere Wiederherstellung im Fehlerfall, Flusskontrolle und Zuverlässigkeit
- Ein verbindungsorientiertes Protokoll
- Lässt sich sehr flexibel mit Fremdsystemen einsetzen, die nur TCP unterstützen
- Routing-fähig
- Es gelten nur statische Datenlängen.
- Meldungen werden quittiert.
- Anwendungen werden über Portnummern adressiert.
- Die meisten Protokolle von Benutzeranwendungen wie TELNET und FTP nutzen TCP.
- Für die Datenverwaltung ist wegen der Programmierschnittstelle SEND/RECEIVE Programmieraufwand erforderlich.

International Standards Organization (ISO) on Transport Control Protocol (TCP) (RFC 1006) (ISO on TCP) ist ein Verfahren, das es ermöglicht, ISO-Anwendungen ins TCP/IP-Netzwerk zu übernehmen. Dieses Protokoll hat die folgenden Merkmale:

- Eng mit der Hardware verknüpftes effizientes Kommunikationsprotokoll
- Geeignet für mittelgroße bis große Datenmengen (bis 8192 Byte)
- Im Gegensatz zu TCP sind die Meldungsfunktion und die Datenendeerkennung meldungsorientiert.
- Routing-fähig, Einsatz in WAN möglich
- Dynamische Datenlängen sind möglich.
- Für die Datenverwaltung ist wegen der Programmierschnittstelle SEND/RECEIVE Programmieraufwand erforderlich.

Mit Transport Service Access Points (TSAPs) gestattet das TCP-Protokoll mehrere Verbindungen mit einer einzigen IP-Adresse (bis zu 64-K-Verbindungen). Bei RFC 1006 ermitteln TSAPs die eindeutige Zuordnung dieser Verbindungen der Kommunikationsendpunkte zu einer IP-Adresse.

#### TSEND\_C und TRCV\_C

Die Anweisung TSEND\_C verbindet die Funktionen der Anweisungen TCON, TDISCON und TSEND. Die Anweisung TRCV\_C verbindet die Funktionen der Anweisungen TCON, TDISCON und TRCV. (Weitere Informationen zu diesen Anweisungen finden Sie unter "TCON, TDISCON, TSEND UND TRCV (Seite 466)".)

Die Mindestdatengröße, die gesendet (TSEND\_C) oder empfangen (TRCV\_C) werden kann, ist ein Byte. Die Maximalgröße beträgt 8192 Byte. TSEND\_C unterstützt nicht die Übertragung von Daten von Booleschen Adressen und TRCV\_C empfängt Daten nicht in Booleschen Adressen. Beachten Sie für weitere Informationen zum Übertragen von Daten mit diesen Anweisungen den Abschnitt zur Datenkonsistenz (Seite 163).

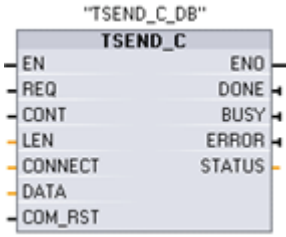
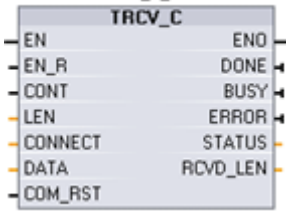
**Hinweis**

**Initialisierung der Kommunikationsparameter**

Nachdem Sie die Anweisung TSEND\_C oder TRCV\_C eingefügt haben, konfigurieren Sie in den "Eigenschaften" der Anweisung (Seite 138) die Kommunikationsparameter. Wenn Sie die Parameter für die Kommunikationspartner im Inspektorfenster eingeben, gibt STEP 7 die entsprechenden Daten in den DB der Anweisung ein.

Wenn Sie einen Multiinstanz-DB verwenden möchten, müssen Sie den DB manuell auf beiden CPUs konfigurieren.

Tabelle 10- 2 Anweisungen TSEND\_C und TRCV\_C

KOP/FUP	SCL	Beschreibung
	<pre>"TSEND_C_DB" (     req:=_bool_in_,     cont:=_bool_in_,     len:=_uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     connect:=_struct_inout_,     data:=_variant_inout_,     com_rst:=_bool_inout_);</pre>	<p>TSEND_C stellt eine TCP- oder ISO-on-TCP-Verbindung zu einem Partner her, sendet Daten und kann die Verbindung auch wieder beenden. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.</p>
	<pre>"TRCV_C_DB" (     en_r:=_bool_in_,     cont:=_bool_in_,     len:=_uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     rcvd_len=&gt;_uint_out_,     connect:=_struct_inout_,     data:=_variant_inout_,     com_rst:=_bool_inout_);</pre>	<p>TRCV_C stellt eine TCP- oder ISO on TCP-Verbindung zu einer CPU her, empfängt Daten und kann die Verbindung auch wieder beenden. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.</p>

<sup>1</sup> STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 10- 3 Datentypen für die Parameter von TSEND\_C und TRCV\_C

Parameter und Datentyp		Datentyp	Beschreibung
REQ (TSEND_C)	IN	Bool	Steuerparameter REQ startet den Sendeauftrag bei steigender Flanke über die in CONNECT beschriebene Verbindung.
EN_R (TRCV_C)	IN	Bool	Steuerparameter freigegeben für Empfang: Ist EN_R = 1, ist TRCV_C empfangsbereit. Der Empfangsauftrag wird bearbeitet.
CONT	IN	Bool	<ul style="list-style-type: none"> <li>0: Verbindung trennen</li> <li>1: Verbindung herstellen und halten</li> </ul>
LEN	IN	UInt	<p>Maximale Bytezahl, die gesendet (TSEND_C) oder empfangen (TRCV_C) werden soll:</p> <ul style="list-style-type: none"> <li>Standard = 0: Der Parameter DATA legt die Länge der zu sendenden (TSEND_C) oder zu empfangenden (TRCV_C) Daten fest.</li> <li>Ad-hoc-Modus = 65535: Für den Datenempfang ist eine variable Länge eingestellt (TRCV_C).</li> </ul>
CONNECT	IN_OUT	TCON_Param	Pointer auf die Verbindungsbeschreibung
DATA	IN_OUT	Variant	<ul style="list-style-type: none"> <li>Enthält Adresse und Länge der Daten, die gesendet (TSEND_C) werden sollen.</li> <li>Enthält Anfangsadresse und maximale Länge der empfangenen Daten (TRCV_C).</li> </ul>
COM_RST	IN_OUT	Bool	<p>Ermöglicht den Neustart der Anweisung:</p> <ul style="list-style-type: none"> <li>0: Irrelevant</li> <li>1: Kompletter Neustart des Funktionsbausteins, bestehende Verbindung wird beendet.</li> </ul>
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0: Auftrag noch nicht gestartet oder läuft noch.</li> <li>1: Auftrag fehlerfrei ausgeführt.</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0: Auftrag beendet.</li> <li>1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.</li> </ul>
ERROR	OUT	Bool	<p>Statusparameter mit den folgenden Werten:</p> <ul style="list-style-type: none"> <li>0: Kein Fehler</li> <li>1: Ein Fehler ist während der Bearbeitung aufgetreten. STATUS enthält ausführliche Informationen über die Fehlerart.</li> </ul>
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Fehler- und Statusparameter" unten.)
RCVD_LEN (TRCV_C)	OUT	Int	Tatsächliche empfangene Datenmenge, in Bytes

---

#### Hinweis

Die Anweisung TSEND\_C erfordert einen Wechsel von 0 nach 1 am Eingangsparameter REQ, um einen Sendeauftrag zu starten. Der Parameter BUSY wird dann während der Verarbeitung auf 1 gesetzt. Die Fertigstellung des Sendeauftrags wird kenntlich gemacht, indem einer der Parameter DONE oder ERROR einen Zyklus lang auf 1 gesetzt wird. Während dieses Zeitraums wird ein Wechsel von 0 nach 1 am Eingangsparameter REQ ignoriert.

---

#### Hinweis

Die Standardeinstellung des Parameters LEN (LEN = 0) ermittelt anhand des Parameters DATA die Länge der zu übertragenden Daten. Stellen Sie sicher, dass die von der Anweisung TSEND\_C gesendeten Daten (DATA) dieselbe Größe haben wie der Parameter DATA der Anweisung TRCV\_C.

---

### Funktionsweise von TSEND\_C

Die folgenden Funktionen beschreiben die Funktionsweise der Anweisung TSEND\_C:

- Um eine Verbindung herzustellen, ist TSEND\_C mit CONT = 1 auszuführen.
- Nach der erfolgreichen Herstellung der Verbindung setzt TSEND\_C den Parameter DONE für einen Zyklus.
- Um die Verbindung zu beenden, führen Sie TSEND\_C mit CONT = 0 aus. Die Verbindung wird dann sofort abgebrochen. Dies betrifft auch den Empfänger. Die Verbindung wird beendet und es können Daten im Empfangspuffer verlorengehen.
- Um Daten über eine bestehende Verbindung zu senden, muss TSEND\_C bei einer steigenden Flanke an REQ ausgeführt werden. Nach dem erfolgreichen Senden setzt TSEND\_C den Parameter DONE für einen Zyklus auf 1.
- Um eine Verbindung herzustellen und Daten zu senden, muss TSEND\_C mit CONT = 1 und REQ = 1 ausgeführt werden. Nach dem erfolgreichen Senden setzt TSEND\_C den Parameter DONE für einen Zyklus auf 1.

### Funktionsweise von TRCV\_C

Die folgenden Funktionen beschreiben die Funktionsweise der Anweisung TRCV\_C:

- Um eine Verbindung herzustellen, ist TRCV\_C mit CONT = 1 auszuführen.
- Um Daten zu empfangen, führen Sie TRCV\_C mit dem Parameter EN\_R = 1 aus. TRCV\_C empfängt kontinuierlich Daten, wenn die Parameter EN\_R = 1 und CONT = 1 sind.
- Um die Verbindung zu beenden, führen Sie TRCV\_C mit dem Parameter CONT = 0 aus. Die Verbindung wird sofort beendet, es können dabei Daten verloren gehen.

TRCV\_C arbeitet mit den gleichen Empfangsmodi wie die Anweisung TRCV. Die folgende Tabelle zeigt, wie Daten in den Empfangsbereich geschrieben werden.

Tabelle 10- 4 Eintragen der Daten in den Empfangsbereich

Protokollvariante	Eintragen der Daten in den Empfangsbereich	Parameter "connection_type"	Wert des Parameters LEN	Wert des Parameters RCVD_LEN (Byte)
TCP	Ad-hoc-Modus	B#16#11	65535	1 bis 1472
TCP	Datenempfang mit angegebener Länge	B#16#11	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192
ISO on TCP	Ad-hoc-Modus	B#16#12	65535	1 bis 1472
ISO on TCP	Protokollgesteuert	B#16#12	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192

---

### Hinweis

#### Ad-hoc-Modus

Den Ad-hoc-Modus gibt es nur bei den Protokollvarianten TCP und ISO on TCP. Den Ad-hoc-Modus stellen Sie ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen. Der Empfangsbereich ist identisch mit dem von Parameter DATA angegebenen Bereich. Die Länge der empfangenen Daten wird am Parameter RCVD\_LEN ausgegeben.

Wenn Sie die Daten in einem "optimierten" DB (nur symbolisch adressierbar) speichern, können Sie Daten nur in Arrays der Datentypen Byte, Char, USInt und SInt empfangen.

---

### Hinweis

#### STEP 7-Projekte der S7-300/400 mit Ad-hoc-Modus in die S7-1200 importieren

In STEP 7-Projekten der S7-300/400 wählen Sie den Ad-hoc-Modus aus, indem Sie dem Parameter LEN den Wert 0 zuweisen. Bei der S7-1200 stellen Sie den Ad-hoc-Modus ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen.

Wenn Sie ein STEP 7-Projekt der S7-300/400, das den Ad-hoc-Modus umfasst, in die S7-1200 importieren, müssen Sie den Parameter LEN in 65535 ändern.

---

### Hinweis

Durch die asynchrone Bearbeitung von TSEND\_C müssen die Daten im Sendebereich konsistent bleiben, bis einer der Parameter DONE oder ERROR den Wert WAHR annimmt.

Für TSEND\_C bedeutet der Status WAHR des Parameters DONE, dass die Daten erfolgreich gesendet wurden. Er bedeutet nicht, dass die Partner-CPU den Empfangspuffer tatsächlich liest.

Durch die asynchrone Bearbeitung von TRCV\_C sind die Daten im Empfangsbereich nur konsistent, wenn der Parameter DONE = 1 ist.

---

Tabelle 10- 5 Parameter BUSY, DONE und ERROR der Anweisungen TSEND\_C und TRCV\_C

BUSY	DONE	ERROR	Beschreibung
TRUE	Irrelevant	Irrelevant	Auftrag wird bearbeitet.
FALSE	TRUE	FALSE	Auftrag erfolgreich ausgeführt.
FALSE	FALSE	TRUE	Auftrag mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS hinterlegt.
FALSCH	FALSCH	FALSCH	Kein neuer Auftrag zugewiesen.

### Parameter Error und Status

Tabelle 10- 6 Bedingungscode von TSEND\_C und TRCV\_C für ERROR und STATUS

ERROR	STATUS	Beschreibung
0	0000	Auftrag fehlerfrei ausgeführt
0	7000	Keine laufende Auftragsbearbeitung
0	7001	Auftragsbearbeitung starten, Verbindung herstellen, auf Verbindungspartner warten
0	7002	Daten werden gesendet oder empfangen
0	7003	Verbindung wird beendet
0	7004	Verbindung hergestellt, wird überwacht, keine laufende Auftragsbearbeitung
1	8085	Der Parameter LEN ist größer als der größte zulässige Wert.
1	8086	Der Parameter CONNECT liegt außerhalb des zulässigen Bereichs.
1	8087	Maximale Anzahl Verbindungen erreicht; keine weitere Verbindung möglich.
1	8088	LEN-Parameter gilt nicht für den in DATA angegebenen Speicherbereich.
1	8089	Der Parameter CONNECT verweist nicht auf einen Datenbaustein.
1	8091	Maximale Schachteltiefe überschritten.
1	809A	Der Parameter CONNECT zeigt auf ein Feld, das nicht mit der Länge der Verbindungsbeschreibung übereinstimmt.
1	809B	local_device_id in der Verbindungsbeschreibung stimmt nicht mit der CPU überein.
1	80A1	Kommunikationsfehler: <ul style="list-style-type: none"> <li>• Die angegebene Verbindung ist noch nicht hergestellt</li> <li>• Die angegebene Verbindung wird derzeit beendet; Datenübertragung über diese Verbindung nicht möglich</li> <li>• Schnittstelle wird neu initialisiert</li> </ul>
1	80A3	Es wird versucht, eine nicht vorhandene Verbindung zu beenden
1	80A4	Die IP-Adresse des entfernten Teilnehmers ist ungültig. Beispiel: Die IP-Adresse des entfernten Teilnehmers ist dieselbe wie die IP-Adresse des lokalen Teilnehmers.
1	80A5	Verbindungs-ID wird bereits verwendet.
1	80A7	Kommunikationsfehler: Sie haben TDISCON aufgerufen, bevor TSEND_C beendet war.
1	80B2	Der Parameter CONNECT zeigt auf einen Datenbaustein, der mit dem Schlüsselwort UNLINKED erzeugt wurde.



ERROR	STATUS	Beschreibung
1	80B3	Inkonsistente Parameter: <ul style="list-style-type: none"> <li>• Fehler in der Verbindungsbeschreibung</li> <li>• Die lokale Schnittstelle (Parameter local_tsap_id) ist bereits in einer anderen Verbindungsbeschreibung vorhanden.</li> <li>• Die ID in der Verbindungsbeschreibung stimmt nicht mit der als Parameter angegebenen ID überein.</li> </ul>
1	80B4	Wenn Sie eine passive Verbindung über ISO on TCP (connection_type = B#16#12) aufbauen, werden Sie anhand des Fehlercodes 80B4 gewarnt, wenn der eingegebene TSAP keiner der folgenden Voraussetzungen für Adressen entspricht: <ul style="list-style-type: none"> <li>• Hat der lokale TSAP eine Länge von 2 und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein.</li> <li>• Hat der lokale TSAP eine Länge von 3 und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein und alle Bytes müssen gültige ASCII-Zeichen sein.</li> <li>• Hat der lokale TSAP eine Länge von 3 oder größer und das erste Byte des ID-Werts des TSAP keinen Wert von E0 oder E1 (hexadezimal), müssen alle Bytes der TSAP-ID gültige ASCII-Zeichen sein.</li> </ul> Gültige ASCII-Zeichen sind Bytewerte von 20 bis 7E (hexadezimal).
1	80B7	Der Datentyp und/oder die Länge der übertragenen Daten entsprechen nicht dem Bereich in der Partner-CPU, in die die Daten geschrieben werden sollen.
1	80C3	Alle Verbindungsressourcen sind belegt.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> <li>• Die Verbindung kann zur Zeit nicht hergestellt werden</li> <li>• Die Schnittstelle empfängt neue Parameter</li> <li>• Die konfigurierte Verbindung wird von TDISCON entfernt.</li> </ul>
1	8722	Parameter CONNECT: Ungültiger Quellbereich: Der Bereich ist im DB nicht vorhanden.
1	873A	Parameter CONNECT: Der Zugriff auf die Verbindungsbeschreibung ist nicht möglich (z. B. DB nicht verfügbar)
1	877F	Parameter CONNECT: Interner Fehler, z. B. ungültiger Verweis auf ANY.
1	893A	Parameter enthält die Nummer eines DB, der nicht geladen ist.

## Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TSEND\_C und TRCV\_C sowie TSEND und TRCV unterstützen alle die Ethernet-Protokolle TCP und ISO on TCP.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 138)".

## Siehe auch

Parameter für die PROFINET-Verbindung (Seite 140)

## TCON, TDISCON, TSEND UND TRCV

### Ethernet-Kommunikation mit den Protokollen TCP und ISO on TCP

#### Hinweis

#### Anweisungen TSEND\_C und TRCV\_C

Um die Programmierung der PROFINET/Ethernet-Kommunikation zu vereinfachen, verbinden die Anweisungen TSEND\_C und TRCV\_C die Funktionalität der Anweisungen TCON, TDISCON, TSEND und TRCV:

- TSEND\_C verbindet die Anweisungen TCON, TDISCON und TSEND.
- TRCV\_C verbindet die Anweisungen TCON, TDISCON und TRCV.

Die folgenden Anweisungen steuern den Kommunikationsprozess:

- TCON baut die TCP/IP-Verbindung zwischen dem Client- und dem Server-PC (CPU) auf.
- TSEND und TRCV senden und empfangen Daten.
- TDISCON trennt die Verbindung.

Die Mindestdatengröße, die gesendet (TSEND) oder empfangen (TRCV) werden kann, ist ein Byte. Die Maximalgröße beträgt 8192 Byte. TSEND unterstützt nicht die Übertragung von Daten von Booleschen Adressen und TRCV empfängt Daten nicht in Booleschen Adressen. Beachten Sie für weitere Informationen zum Übertragen von Daten mit diesen Anweisungen den Abschnitt zur Datenkonsistenz (Seite 163).

TCON, TDISCON, TSEND und TRCV laufen asynchron ab, so dass die Bearbeitung eines Auftrags mehrere Anweisungsausführungen umfasst. Beispielsweise starten Sie einen Auftrag zum Einrichten und Herstellen einer Verbindung durch die Ausführung von TCON mit Parameter REQ = 1. Dann überwachen Sie den Stand der Auftragsausführung, indem TCON wiederholt ausgeführt wird, und prüfen, ob Parameter DONE den Abschluss des Auftrags bestätigt.

Die folgende Tabelle zeigt die Beziehungen zwischen BUSY, DONE und ERROR. Den aktuellen Status eines Auftrags können Sie mit der Tabelle feststellen.

Tabelle 10- 7 Interaktionen zwischen den Parametern BUSY, DONE und ERROR

BUSY	DONE	ERROR	Beschreibung
TRUE	Irrelevant	Irrelevant	Auftrag wird bearbeitet.
FALSE	TRUE	FALSE	Auftrag erfolgreich ausgeführt.
FALSE	FALSE	TRUE	Auftrag mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS hinterlegt.
FALSCH	FALSCH	FALSCH	Kein neuer Auftrag zugewiesen.

## TCON und TDISCON

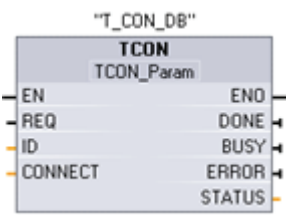
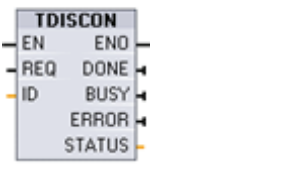
### Hinweis

#### Initialisierung der Kommunikationsparameter

Nachdem Sie die Anweisung TCON eingefügt haben, konfigurieren Sie in den "Eigenschaften" der Anweisung (Seite 138) die Kommunikationsparameter. Wenn Sie die Parameter für die Kommunikationspartner im Inspektorfenster eingeben, gibt STEP 7 die entsprechenden Daten in den Instanz-DB der Anweisung ein.

Wenn Sie einen Multiinstanz-DB verwenden möchten, müssen Sie den DB manuell auf beiden CPUs konfigurieren.

Tabelle 10- 8 Anweisungen TCON und TDISCON

KOP/FUP		Beschreibung
	<pre>"TCON_DB" (   req:=_bool_in_,   ID:=_undef_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   connect:=_struct_inout_);</pre>	TCP und ISO on TCP: TCON initiiert eine Kommunikationsverbindung von der CPU zu einem Kommunikationsteilnehmer.
	<pre>"TDISCON_DB" (   req:=_bool_in_,   ID:=_word_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_);</pre>	TCP und ISO on TCP: TDISCON beendet eine Kommunikationsverbindung von der CPU zu einem Kommunikationsteilnehmer.

<sup>1</sup> STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 10- 9 Datentypen für die Parameter von TCON und TDISCON

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
ID	IN	CONN_OUC (Word)

Steuerparameter REQ startet den Auftrag zur Herstellung der von ID angegebenen Verbindung. Der Auftrag beginnt mit einer steigenden Flanke.

Verweis auf die Verbindung, die mit dem entfernten Teilnehmer oder zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems hergestellt (TCON) oder beendet (TDISCON) werden soll. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen.  
Wertebereich: W#16#0001 bis W#16#0FFF

Parameter und Datentyp		Datentyp	Beschreibung
CONNECT (TCON)	IN_OUT	TCON_Param	Pointer auf die Verbindungsbeschreibung
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0: Auftrag noch nicht gestartet oder läuft noch.</li> <li>1: Auftrag fehlerfrei ausgeführt.</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0: Auftrag beendet.</li> <li>1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.</li> </ul>
ERROR	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> <li>0: Kein Fehler</li> <li>1: Ein Fehler ist während der Bearbeitung aufgetreten. STATUS enthält ausführliche Informationen über die Fehlerart.</li> </ul>
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungscode für Error und Status" unten.)

Beide Kommunikationsteilnehmer führen die Anweisung TCON aus, um die Kommunikationsverbindung einzurichten und herzustellen. Mit Parametern geben Sie die aktiven und passiven Endpunkte der Kommunikation an. Nach dem Einrichten und Aufbauen der Verbindung wird sie automatisch von der CPU gehalten und überwacht.

Wird die Verbindung durch eine Leitungsstörung oder vom entfernten Teilnehmer abgebrochen, so versucht der aktive Teilnehmer, sie wieder aufzubauen. Sie müssen TCON nicht erneut ausführen.

Wird die Anweisung TDISCON ausgeführt oder ist die CPU in den Betriebszustand STOP gegangen, so wird die bestehende Verbindung beendet und die eingerichtete Verbindung gelöscht. Um die Verbindung einzurichten und wiederherzustellen, müssen Sie TCON erneut ausführen.

Tabelle 10- 10 Bedingungscode von ERROR und STATUS für TCON und TDISCON

ERROR	STATUS	Beschreibung
0	0000	Verbindung wurde erfolgreich aufgebaut.
0	7000	Keine laufende Auftragsbearbeitung
0	7001	Auftragsbearbeitung starten, Verbindung herstellen (TCON) oder Verbindung beenden (TDISCON)
0	7002	Folgeaufruf (REQ irrelevant), Verbindung herstellen (TCON) oder Verbindung beenden (TDISCON)
1	8086	Parameter ID ist außerhalb des zulässigen Adressbereichs.
1	8087	TCON: Maximale Anzahl Verbindungen erreicht; keine weitere Verbindung möglich.
1	809B	TCON: local_device_id in der Verbindungsbeschreibung stimmt nicht mit der CPU überein.
1	80A1	TCON: Verbindung oder Port ist vom Anwender belegt.
1	80A2	TCON: Lokaler oder entfernter Port ist vom System belegt.
1	80A3	Es wird versucht, eine vorhandene Verbindung erneut aufzubauen (TCON) oder eine nicht vorhandene Verbindung zu beenden (TDISCON).

ERROR	STATUS	Beschreibung
1	80A4	TCON: IP-Adresse des entfernten Verbindungspunkts ist ungültig, sie entspricht möglicherweise der lokalen IP-Adresse.
1	80A5	TCON: Verbindungs-ID wird bereits verwendet.
1 ( )	80A7	TCON: Kommunikationsfehler: Sie haben TDISCON ausgeführt, bevor TCON beendet war. TDISCON muss zunächst die von ID angegebene Verbindung vollständig beenden.
1	80B4	TCON: Wenn Sie eine passive Verbindung über ISO on TCP (connection_type = B#16#12) aufbauen, werden Sie anhand des Fehlercodes 80B4 gewarnt, wenn der eingegebene TSAP keiner der folgenden Voraussetzungen für Adressen entspricht: <ul style="list-style-type: none"> <li>• Hat der lokale TSAP eine Länge von 2 und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein.</li> <li>• Hat der lokale TSAP eine Länge von 3 und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein und alle Bytes müssen gültige ASCII-Zeichen sein.</li> <li>• Hat der lokale TSAP eine Länge von 3 oder größer und das erste Byte des ID-Werts des TSAP keinen Wert von E0 oder E1 (hexadezimal), müssen alle Bytes der TSAP-ID gültige ASCII-Zeichen sein.</li> </ul> Gültige ASCII-Zeichen sind Bytewerte von 20 bis 7E (hexadezimal).
1	80B6	TCON: Parametrierungsfehler in Parameter connection_type
1	80B7	TCON: Datentyp und/oder Länge der übertragenen Daten entsprechen nicht dem Bereich in der Partner-CPU, in den die Daten geschrieben werden sollen.
1)	80B8	TCON: Parameter der lokalen Verbindungsbeschreibung und Parameter-ID sind verschieden.
1	80C3	TCON: Alle Verbindungsressourcen sind belegt.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> <li>• Die Verbindung kann zur Zeit nicht hergestellt werden (TCON).</li> <li>• Die konfigurierte Verbindung wird von TDISCON (TCON) entfernt.</li> <li>• Die angegebene Verbindung wird derzeit aufgebaut (TDISCON).</li> <li>• Die Schnittstelle empfängt neue Parameter (TCON und TDISCON).</li> </ul>

TSEND und TRCV

Tabelle 10- 11 Anweisungen TSEND und TRCV

KOP/FUP	SCL	Beschreibung
<p>"T_SEND_DB"</p>	<pre>"TSEND_DB" (     req:= _bool_in_,     ID:= _word_in_,     len:= _uint_in_,     done=&gt; _bool_out_,     busy=&gt; _bool_out_,     error=&gt; _bool_out_,     status=&gt; _word_out_,     data:= _variant_inout_ );</pre>	<p>TCP und ISO on TCP: TSEND sendet über eine Kommunikationsverbindung Daten von der CPU zu einer Partnerstation.</p>
<p>"T_RCV_DB"</p>	<pre>"TRCV_DB" (     en_r:= _bool_in_,     ID:= _word_in_,     len:= _uint_in_,     ndr=&gt; _bool_out_,     busy=&gt; _bool_out_,     error=&gt; _bool_out_,     status=&gt; _word_out_,     rcvd_len=&gt; _uint_out_,     data:= _variant_inout_ );</pre>	<p>TCP und ISO on TCP: TRCV empfängt über eine Kommunikationsverbindung Daten von einer Partnerstation auf der CPU.</p>

1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 10- 12 Datentypen für die Parameter von TSEND und TRCV

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
EN_R	IN	Bool
ID	IN	CONN_OUC (Word)
LEN	IN	UInt

TSEND: Startet den Sendeauftrag bei einer steigenden Flanke. Die Daten werden aus dem mit DATA und LEN bezeichneten Bereich übertragen.

TRCV: Aktiviert die CPU für den Empfang, bei EN\_R = 1 ist TRCV empfangsbereit. Der Empfangsauftrag wird bearbeitet.

Verweis auf die zugehörige Verbindung. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen.  
Wertebereich: W#16#0001 bis W#16#0FFF

Maximale Bytezahl, die gesendet (TSEND) oder empfangen (TRCV) werden soll:

- Standard = 0: Der Parameter DATA legt die Länge der zu sendenden (TSEND) oder zu empfangenden (TRCV) Daten fest.
- Ad-hoc-Modus = 65535: Für den Datenempfang ist eine variable Länge eingestellt (TRCV).

Parameter und Datentyp		Datentyp	Beschreibung
DATA	IN_OUT	Variante	Pointer auf den Datenbereich für Senden (TSEND) oder Empfangen (TRCV). Der Datenbereich enthält die Adresse und Länge. Die Adresse bezieht sich auf einen der Speicherbereiche E, A oder M oder auf einen DB.
DONE	OUT	Bool	TSEND: <ul style="list-style-type: none"> <li>• 0: Auftrag noch nicht gestartet oder läuft noch.</li> <li>• 1: Auftrag fehlerfrei ausgeführt.</li> </ul>
NDR	OUT	Bool	TRCV: <ul style="list-style-type: none"> <li>• NDR = 0: Auftrag noch nicht gestartet oder läuft noch.</li> <li>• NDR = 1: Auftrag erfolgreich beendet.</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• BUSY = 1: Der Auftrag ist noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.</li> <li>• BUSY = 0: Auftrag beendet.</li> </ul>
ERROR	OUT	Bool	ERROR = 1: Ein Fehler ist während der Bearbeitung aufgetreten. STATUS enthält ausführliche Informationen über die Fehlerart.
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungscode für Error und Status" unten.)
RCVD_LEN	OUT	Int	TRCV: Tatsächliche empfangene Datenmenge in Byte

#### Hinweis

Die Anweisung TSEND erfordert einen Wechsel von 0 nach 1 am Eingangsparameter REQ, um einen Sendeauftrag zu starten. Der Parameter BUSY wird dann während der Verarbeitung auf 1 gesetzt. Die Fertigstellung des Sendeauftrags wird kenntlich gemacht, indem einer der Parameter DONE oder ERROR einen Zyklus lang auf 1 gesetzt wird. Während dieses Zeitraums wird ein Wechsel von 0 nach 1 am Eingangsparameter REQ ignoriert.

#### Funktionsweise von TRCV

Die Anweisung TRCV schreibt die empfangenen Daten in einen Empfangsbereich, der von den folgenden zwei Variablen angegeben wird:

- Zeigt auf den Anfang des Bereichs
- Länge des Bereichs oder der Wert, der am Eingang LEN bereitgestellt wird, sofern er nicht 0 ist

#### Hinweis

Die Standardeinstellung des Parameters LEN (LEN = 0) ermittelt anhand des Parameters DATA die Länge der zu übertragenden Daten. Stellen Sie sicher, dass die von der Anweisung TSEND gesendeten Daten (DATA) dieselbe Größe haben wie der Parameter DATA der Anweisung TRCV.

Sobald alle Auftragsdaten empfangen sind, werden sie von TRCV in den Empfangsbereich übertragen und NDR wird auf 1 gesetzt.

Tabelle 10- 13 Eintragen der Daten in den Empfangsbereich

Protokollvariante	Eintragen der Daten in den Empfangsbereich	Parameter "connection_type"	Wert des Parameters LEN	Wert des Parameters RCVD_LEN (Byte)
TCP	Ad-hoc-Modus	B#16#11	65535	1 bis 1472
TCP	Datenempfang mit angegebener Länge	B#16#11	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192
ISO on TCP	Ad-hoc-Modus	B#16#12	65535	1 bis 1472
ISO on TCP	protokollgesteuert	B#16#12	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192

**Hinweis**

**Ad-hoc-Modus**

Den Ad-hoc-Modus gibt es nur bei den Protokollvarianten TCP und ISO on TCP. Den Ad-hoc-Modus stellen Sie ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen. Der Empfangsbereich ist identisch mit dem von Parameter DATA angegebenen Bereich. Die Länge der empfangenen Daten wird am Parameter RCVD\_LEN ausgegeben. Unmittelbar nach dem Empfang eines Datenblocks schreibt TRCV die Daten in den Empfangsbereich und setzt NDR auf 1.

Wenn Sie die Daten in einem "optimierten" DB (nur symbolisch adressierbar) speichern, können Sie Daten nur in Arrays der Datentypen Byte, Char, USInt und SInt empfangen.

**Hinweis**

**STEP 7-Projekte der S7-300/400 mit Ad-hoc-Modus in die S7-1200 importieren**

In STEP 7-Projekten der S7-300/400 wählen Sie den Ad-hoc-Modus aus, indem Sie dem Parameter LEN den Wert 0 zuweisen. Bei der S7-1200 stellen Sie den Ad-hoc-Modus ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen.

Wenn Sie ein STEP 7-Projekt der S7-300/400, das den Ad-hoc-Modus umfasst, in die S7-1200 importieren, müssen Sie den Parameter LEN in 65535 ändern.

Tabelle 10- 14 Bedingungscode von ERROR und STATUS für TSEND und TRCV

ERROR	STATUS	Beschreibung
0	0000	<ul style="list-style-type: none"> <li>• Sendeauftrag fehlerfrei ausgeführt (TSEND)</li> <li>• Neue Daten angenommen: Die aktuelle Länge der empfangenen Daten wird in RCVD_LEN (TRCV) gezeigt.</li> </ul>
0	7000	<ul style="list-style-type: none"> <li>• Keine laufende Auftragsbearbeitung (TSEND)</li> <li>• Baustein nicht empfangsbereit (TRCV)</li> </ul>



ERROR	STATUS	Beschreibung
0	7001	<ul style="list-style-type: none"> <li>• Auftragsbearbeitung starten, Daten werden gesendet: Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu (TSEND).</li> <li>• Baustein empfangsbereit, Empfangsauftrag wurde aktiviert (TRCV).</li> </ul>
0	7002	<ul style="list-style-type: none"> <li>• Folgeaufruf (REQ irrelevant), Auftrag wird bearbeitet: Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu (TSEND).</li> <li>• Folgeaufruf, Empfangsauftrag wird bearbeitet: Während dieser Bearbeitung werden die Daten in den Empfangsbereich geschrieben. Deshalb kann ein Fehler zu inkonsistenten Daten im Empfangsbereich führen (TRCV).</li> </ul>
1	8085	<ul style="list-style-type: none"> <li>• Der Parameter LEN ist größer als der größte zulässige Wert (TSEND) und (TRCV).</li> <li>• Der Parameter LEN oder DATA hat sich seit der ersten Ausführung der Anweisung geändert (TRCV).</li> </ul>
1	8086	Der Parameter ID ist nicht im zulässigen Adressbereich.
1	8088	Der Parameter LEN ist größer als der in DATA festgelegte Speicherbereich.
1	80A1	Kommunikationsfehler: <ul style="list-style-type: none"> <li>• Die angegebene Verbindung ist noch nicht hergestellt (TSEND und TRCV).</li> <li>• Die angegebene Verbindung wird beendet. Ein Sende- oder Empfangsauftrag ist über diese Verbindung nicht möglich (TSEND und TRCV).</li> <li>• Die Schnittstelle wird neu initialisiert (TSEND).</li> <li>• Die Schnittstelle empfängt neue Parameter (TRCV).</li> </ul>
1	80C3	Interner Mangel an Ressourcen: Ein Baustein mit dieser ID wird bereits in einer anderen Prioritätsklasse bearbeitet.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> <li>• Die Verbindung zum Kommunikationspartner kann zur Zeit nicht hergestellt werden.</li> <li>• Die Schnittstelle empfängt neue Parametrierungen oder die Verbindung wird gerade aufgebaut.</li> </ul>

### Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TSEND\_C, TRCV\_C, TSEND und TRCV unterstützen alle die Ethernet-Protokolle TCP und ISO on TCP.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 138)".

### Siehe auch

Parameter für die PROFINET-Verbindung (Seite 140)

### 10.2.2.5 UDP

UDP ist ein Standardprotokoll, beschrieben von RFC 768: User Datagram Protocol. UDP bietet ein Verfahren, damit eine Anwendung ein Datengramm zu einer anderen Anwendung senden kann. Die Zustellung der Daten ist jedoch nicht garantiert. Dieses Protokoll hat die folgenden Merkmale:

- Schnelles Kommunikationsprotokoll, weil eng mit der Hardware verbunden
- Geeignet für kleine bis mittelgroße Datenmengen (bis 2048 Byte)
- UDP ist ein einfacheres Transportprotokoll als TCP, mit einer dünnen Schicht und wenig Verwaltungsaufwand
- Lässt sich sehr flexibel in vielen Fremdsystemen einsetzen
- Routing-fähig
- Leitet die Datagramme über Portnummern
- Meldungen werden nicht quittiert: Die Anwendung hat die Verantwortung für Wiederherstellung im Fehlerfall und für Sicherheit zu übernehmen
- Für die Datenverwaltung ist aufgrund der Programmierschnittstelle SEND/RECEIVE Programmieraufwand erforderlich

UDP unterstützt die Broadcast-Kommunikation. Um die Broadcast-Funktion zu nutzen, konfigurieren Sie den entsprechenden Teil der IP-Adresse in der ADDR-Konfiguration. Beispiel: Eine CPU mit der IP-Adresse 192.168.2.10 und der Subnetzmaske 255.255.255.0 hätte die Broadcast-Adresse 192.168.2.255.

### TUSEND und TURCV

Die folgenden Anweisungen steuern den UDP-Kommunikationsprozess:

- TCON baut die Kommunikation zwischen dem Client- und dem Server-PC (CPU) auf.
- TUSEND und TURCV senden und empfangen Daten.
- TDISCON trennt die Kommunikation zwischen dem Client und dem Server.

Weitere Informationen zu den Kommunikationsanweisungen TCON und TDISCON finden Sie im Abschnitt "TCP und ISO-on-TCP" unter TCON, TDISCON, TSEND und TRCV (Seite 466).

Tabelle 10- 15 Anweisungen TUSEND und TURCV

KOP/FUP	SCL	Beschreibung
<p>"TUSEND_DB"</p>	<pre>"TUSEND_DB" (     req:= _bool_in_,     ID:= _word_in_,     len:= _uint_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     data:= variant inout );</pre>	<p>Die Anweisung TUSEND sendet Daten über UDP an den entfernten Partner, der vom Parameter ADDR angegeben wird.</p> <p>Um den Auftrag zum Senden von Daten zu starten, rufen Sie die Anweisung TUSEND mit REQ = 1 auf.</p>
<p>"TURCV_DB"</p>	<pre>"TURCV_DB" (     en_r:= _bool_in_,     ID:= _word_in_,     len:= _uint_in_,     ndr=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     rcvd_len=&gt;_uint_out_,     data:= variant inout );</pre>	<p>Die Anweisung TURCV empfängt Daten über UDP. Der Parameter ADDR zeigt die Adresse des Senders an. Nach erfolgreicher Durchführung von TURCV enthält der Parameter ADDR die Adresse des entfernten Partners (des Senders).</p> <p>TURCV unterstützt den Ad-hoc-Modus nicht.</p> <p>Um den Auftrag zum Empfangen von Daten zu starten, rufen Sie die Anweisung TURCV mit EN_R = 1 auf.</p>

<sup>1</sup> STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

TCON, TDISCON, TUSEND und TURCV laufen asynchron ab, so dass die Bearbeitung eines Auftrags mehrere Anweisungsausführungen umfasst.

Tabelle 10- 16 Datentypen für die Parameter von TUSEND und TURCV

Parameter und Datentyp	Datentyp	Beschreibung
REQ (TUSEND)	IN	Bool
EN_R (TURCV)	IN	Bool
ID	IN	Word
LEN	IN	UInt

Startet den Sendeauftrag bei einer steigenden Flanke. Die Daten werden aus dem mit DATA und LEN bezeichneten Bereich übertragen.

- 0: CPU kann nicht empfangen.
- 1: Aktiviert die CPU für den Empfang. Die Anweisung TURCV ist empfangsbereit und der Empfangsauftrag wird bearbeitet.

Verweis auf die zugehörige Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen.  
Wertebereich: W#16#0001 bis W#16#0FFF.

Anzahl der Bytes, die gesendet (TUSEND) oder empfangen (TURCV) werden sollen.

- Standard = 0. Der Parameter DATA legt die Länge der zu sendenden oder zu empfangenden Daten fest.
- Ansonsten, Wertebereich: 1 bis 1472

Parameter und Datentyp		Datentyp	Beschreibung
DONE (TUSEND)	IN	Bool	Statusparameter DONE (TUSEND): <ul style="list-style-type: none"> <li>• 0: Auftrag noch nicht gestartet oder läuft noch.</li> <li>• 1: Auftrag fehlerfrei ausgeführt.</li> </ul>
NDR (TURCV)	OUT	Bool	Statusparameter NDR (TURCV): <ul style="list-style-type: none"> <li>• 0: Auftrag noch nicht gestartet oder läuft noch.</li> <li>• 1: Auftrag wurde erfolgreich ausgeführt.</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.</li> <li>• 0: Auftrag ist beendet.</li> </ul>
ERROR	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> <li>• 0: Kein Fehler</li> <li>• 1: Ein Fehler ist während der Bearbeitung aufgetreten. STATUS enthält ausführliche Informationen über die Fehlerart.</li> </ul>
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungs-codes für Error und Status" unten.)
DATA	IN_OUT	Variant	Adresse des Sendebereichs (TUSEND) oder Empfangsbereichs (TURCV): <ul style="list-style-type: none"> <li>• das Prozessabbild der Eingänge</li> <li>• das Prozessabbild der Ausgänge</li> <li>• ein Merkerbit</li> <li>• einen Datenbaustein</li> </ul>
ADDR	IN_OUT	Variant	Pointer auf die Adresse des Empfängers (bei TUSEND) oder des Senders (bei TURCV) (z. B. P#DB100.DBX0.0 Byte 8) Der Pointer kann auf einen beliebigen Speicherbereich verweisen. Erforderlich ist eine Struktur von 8 Byte wie folgt: <ul style="list-style-type: none"> <li>• Die ersten 4 Byte enthalten die dezentrale IP-Adresse.</li> <li>• Die nächsten 2 Byte geben die dezentrale Portnummer an.</li> <li>• Die letzten 2 Byte sind reserviert.</li> </ul>

Der Auftragszustand wird an den Ausgangsparametern BUSY und STATUS angezeigt. STATUS entspricht dem Ausgangsparameter RET\_VAL bei asynchron funktionierenden Anweisungen.

Die folgende Tabelle zeigt die Beziehungen zwischen BUSY, DONE (TUSEND), NDR (TURCV) und ERROR. Mithilfe dieser Tabelle können Sie den aktuellen Zustand der Anweisung (TUSEND oder TURCV) ermitteln bzw. den Zeitpunkt, zu dem der Sendevorgang (die Übertragung) bzw. der Empfangsvorgang beendet ist.

Tabelle 10- 17 Zustand der Parameter BUSY, DONE (TUSEND)/NDR (TURCV) und ERROR

BUSY	DONE / NDR	ERROR	Beschreibung
WAHR	Irrelevant	Irrelevant	Auftrag wird bearbeitet.
FALSCH	WAHR	FALSCH	Der Auftrag wurde erfolgreich ausgeführt.

BUSY	DONE / NDR	ERROR	Beschreibung
FALSCH	FALSCH	WAHR	Auftrag mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS hinterlegt.
FALSCH	FALSCH	FALSCH	Der Anweisung wurde kein (neuer) Auftrag zugewiesen.

<sup>1</sup> Wegen der asynchronen Funktion der Anweisungen: Bei TUSENDmüssen die Daten im Sendebereich konsistent bleiben, bis einer der Parameter DONE oder ERROR den Wert WAHR annimmt. Bei TURCVsind die Daten im Empfangsbereich nur konsistent, wenn der Parameter NDR den Wert WAHR hat.

Tabelle 10- 18 Bedingungs-codes von TUSEND und TURCV für ERROR und STATUS

ERROR	STATUS	Beschreibung
0	0000	<ul style="list-style-type: none"> <li>• Sendeauftrag fehlerfrei ausgeführt (TUSEND).</li> <li>• Neue Daten wurden angenommen. Die aktuelle Länge der empfangenen Daten wird in RCVD_LEN (TURCV) gezeigt.</li> </ul>
0	7000	<ul style="list-style-type: none"> <li>• Keine laufende Auftragsbearbeitung (TUSEND)</li> <li>• Baustein nicht empfangsbereit (TURCV)</li> </ul>
0	7001	<ul style="list-style-type: none"> <li>• Auftragsbearbeitung starten, Daten werden gesendet (TUSEND): Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu.</li> <li>• Baustein empfangsbereit, Empfangsauftrag wurde aktiviert (TURCV).</li> </ul>
0	7002	<ul style="list-style-type: none"> <li>• Folgeaufruf (REQ irrelevant), Auftrag wird bearbeitet (TUSEND): Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu.</li> <li>• Folgeaufruf, Auftrag wird bearbeitet: Während der Bearbeitung schreibt die Anweisung TURCV Daten in den Empfangsbereich. Deshalb kann ein Fehler zu inkonsistenten Daten im Empfangsbereich führen.</li> </ul>
1	8085	Der Parameter LEN ist größer als der größte zulässige Wert, hat den Wert 0 (TUSEND) oder Sie haben den Wert des Parameters LEN oder DATA seit der ersten Anweisungsausführung geändert (TURCV).
1	8086	Der Parameter ID ist nicht im zulässigen Adressbereich.
1	8088	<ul style="list-style-type: none"> <li>• Der Parameter LEN ist größer als der in DATA festgelegte Speicherbereich (TUSEND) oder Empfangsbereich (TURCV).</li> <li>• Empfangsbereich ist zu klein (TURCV).</li> </ul>
1	8089	Der Parameter ADDR verweist nicht auf einen Datenbaustein.
1	80A1	Kommunikationsfehler: <ul style="list-style-type: none"> <li>• Die angegebene Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems wurde noch nicht aufgebaut.</li> <li>• Die angegebene Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems wird gerade beendet. Ein Sende- (TUSEND) oder Empfangsauftrag (TURCV) ist über diese Verbindung nicht möglich.</li> <li>• Schnittstelle wird neu initialisiert.</li> </ul>
1	80A4	IP-Adresse des entfernten Verbindungspunkts ist ungültig, sie entspricht möglicherweise der lokalen IP-Adresse (TUSEND).

ERROR	STATUS	Beschreibung
1	80B3	<ul style="list-style-type: none"> <li>Die festgelegte Protokollvariante (Parameter connection_type in der Verbindungsbeschreibung) ist kein UDP. Bitte verwenden Sie die Anweisung TSEND oder TRCV.</li> <li>Parameter ADDR: Ungültige Einstellungen für die Portnummer (TUSEND)</li> </ul>
1	80C3	<ul style="list-style-type: none"> <li>Ein Baustein mit dieser ID wird bereits in einer anderen Prioritätsklasse bearbeitet.</li> <li>Interner Mangel an Ressourcen</li> </ul>
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> <li>Die Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems kann derzeit nicht hergestellt werden (TUSEND).</li> <li>Die Schnittstelle empfängt neue Parameter (TUSEND).</li> <li>Die angegebene Verbindung wird derzeit neu aufgebaut (TURCV).</li> </ul>

### Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TUSEND und TURCV unterstützen das UDP-Ethernet-Protokoll.

Weitere Informationen finden Sie im Kapitel "Gerätekonfiguration" unter "Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren" (Seite 138).

### Betrieb

Beide Partner sind in der UDP-Kommunikation passiv. Typische Startwerte für die Parameter des Datentyps TCON\_Param werden in der folgenden Tabelle gezeigt. Die Portnummern (LOCAL\_TSAP\_ID) sind im 2-Byte-Format geschrieben. Alle Ports mit Ausnahme von 161, 34962, 34963 und 34964 sind zulässig.

Tabelle 10- 19 "Parameterwerte für den Datentyp TCON\_Param

Anweisung TCON	TCON "UDP Conn DB"																																																																																																						
	<table border="1"> <thead> <tr> <th colspan="6">UDP_Conn_DB</th> </tr> <tr> <th></th> <th>Name</th> <th>Datentyp</th> <th>Offset</th> <th>Startwert</th> <th>Kommentar</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>Params</td> <td>TCON_Param</td> <td>0.0</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>BLOCK_LENGTH</td> <td>UInt</td> <td>0.0</td> <td>64</td> <td>byte length of SDT</td> </tr> <tr> <td>4</td> <td>ID</td> <td>CONN_OUC</td> <td>2.0</td> <td>1</td> <td>reference to the connection</td> </tr> <tr> <td>5</td> <td>CONNECTION_TYPE</td> <td>USInt</td> <td>4.0</td> <td>19</td> <td>17: TCP/IP; 18: ISO on TCP</td> </tr> <tr> <td>6</td> <td>ACTIVE_EST</td> <td>Bool</td> <td>5.0</td> <td>false</td> <td>active/passive connection establishment</td> </tr> <tr> <td>7</td> <td>LOCAL_DEVICE_ID</td> <td>USInt</td> <td>6.0</td> <td>1</td> <td>1: local IE interface</td> </tr> <tr> <td>8</td> <td>LOCAL_TSAP_ID_LEN</td> <td>USInt</td> <td>7.0</td> <td>2</td> <td>byte length of local TSAP id/port number</td> </tr> <tr> <td>9</td> <td>REM_SUBNET_ID_LEN</td> <td>USInt</td> <td>8.0</td> <td>0</td> <td>byte length of remote subnet id</td> </tr> <tr> <td>10</td> <td>REM_STADDR_LEN</td> <td>USInt</td> <td>9.0</td> <td>0</td> <td>byte length of remote IP address</td> </tr> <tr> <td>11</td> <td>REM_TSAP_ID_LEN</td> <td>USInt</td> <td>10.0</td> <td>0</td> <td>byte length of remote port/TSAP id</td> </tr> <tr> <td>12</td> <td>NEXT_STADDR_LEN</td> <td>USInt</td> <td>11.0</td> <td>0</td> <td>byte length of next station address</td> </tr> <tr> <td>13</td> <td>LOCAL_TSAP_ID</td> <td>Array[1..16] of Byte</td> <td>12.0</td> <td></td> <td>TSAP id/local port number</td> </tr> <tr> <td>14</td> <td>LOCAL_TSAP_ID[1]</td> <td>Byte</td> <td></td> <td>B#16#07</td> <td></td> </tr> <tr> <td>15</td> <td>LOCAL_TSAP_ID[2]</td> <td>Byte</td> <td></td> <td>B#16#D0</td> <td></td> </tr> </tbody> </table>	UDP_Conn_DB							Name	Datentyp	Offset	Startwert	Kommentar	1	Static					2	Params	TCON_Param	0.0			3	BLOCK_LENGTH	UInt	0.0	64	byte length of SDT	4	ID	CONN_OUC	2.0	1	reference to the connection	5	CONNECTION_TYPE	USInt	4.0	19	17: TCP/IP; 18: ISO on TCP	6	ACTIVE_EST	Bool	5.0	false	active/passive connection establishment	7	LOCAL_DEVICE_ID	USInt	6.0	1	1: local IE interface	8	LOCAL_TSAP_ID_LEN	USInt	7.0	2	byte length of local TSAP id/port number	9	REM_SUBNET_ID_LEN	USInt	8.0	0	byte length of remote subnet id	10	REM_STADDR_LEN	USInt	9.0	0	byte length of remote IP address	11	REM_TSAP_ID_LEN	USInt	10.0	0	byte length of remote port/TSAP id	12	NEXT_STADDR_LEN	USInt	11.0	0	byte length of next station address	13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		TSAP id/local port number	14	LOCAL_TSAP_ID[1]	Byte		B#16#07		15	LOCAL_TSAP_ID[2]	Byte		B#16#D0	
UDP_Conn_DB																																																																																																							
	Name	Datentyp	Offset	Startwert	Kommentar																																																																																																		
1	Static																																																																																																						
2	Params	TCON_Param	0.0																																																																																																				
3	BLOCK_LENGTH	UInt	0.0	64	byte length of SDT																																																																																																		
4	ID	CONN_OUC	2.0	1	reference to the connection																																																																																																		
5	CONNECTION_TYPE	USInt	4.0	19	17: TCP/IP; 18: ISO on TCP																																																																																																		
6	ACTIVE_EST	Bool	5.0	false	active/passive connection establishment																																																																																																		
7	LOCAL_DEVICE_ID	USInt	6.0	1	1: local IE interface																																																																																																		
8	LOCAL_TSAP_ID_LEN	USInt	7.0	2	byte length of local TSAP id/port number																																																																																																		
9	REM_SUBNET_ID_LEN	USInt	8.0	0	byte length of remote subnet id																																																																																																		
10	REM_STADDR_LEN	USInt	9.0	0	byte length of remote IP address																																																																																																		
11	REM_TSAP_ID_LEN	USInt	10.0	0	byte length of remote port/TSAP id																																																																																																		
12	NEXT_STADDR_LEN	USInt	11.0	0	byte length of next station address																																																																																																		
13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		TSAP id/local port number																																																																																																		
14	LOCAL_TSAP_ID[1]	Byte		B#16#07																																																																																																			
15	LOCAL_TSAP_ID[2]	Byte		B#16#D0																																																																																																			

Die Anweisung TUSEND sendet Daten über UDP an den entfernten Partner, der vom Datentyp TADDR\_Param angegeben wird. Die Anweisung TURCV empfängt Daten über UDP. Nach erfolgreicher Ausführung der Anweisung TURCV zeigt der Datentyp TADDR\_Param Ihnen die Adresse des entfernten Partners (des Senders) an.

Tabelle 10- 20 "Parameterwerte für den Datentyp TADDR\_Param

Anweisung TUSEND	TUSEND "UDP ADDR DB"																																																							
	<table border="1"> <thead> <tr> <th colspan="5">Send_UDP_ADDR</th> </tr> <tr> <th>Name</th> <th>Datentyp</th> <th>Offset</th> <th>Startwert</th> <th>Kommentar</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>Addr_Data</td> <td>TADDR_Param</td> <td>0.0</td> <td></td> </tr> <tr> <td>3</td> <td>REM_IP_ADDR</td> <td>Array[1..4] of USint</td> <td>0.0</td> <td>remote station address</td> </tr> <tr> <td>4</td> <td>REM_IP_ADDR[1]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>5</td> <td>REM_IP_ADDR[2]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>6</td> <td>REM_IP_ADDR[3]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>7</td> <td>REM_IP_ADDR[4]</td> <td>USint</td> <td>0</td> <td></td> </tr> <tr> <td>8</td> <td>REM_PORT_NR</td> <td>UInt</td> <td>4.0</td> <td>remote port number</td> </tr> <tr> <td>9</td> <td>RESERVED</td> <td>Word</td> <td>6.0</td> <td>unused; has to be 0</td> </tr> </tbody> </table>	Send_UDP_ADDR					Name	Datentyp	Offset	Startwert	Kommentar	1	Static				2	Addr_Data	TADDR_Param	0.0		3	REM_IP_ADDR	Array[1..4] of USint	0.0	remote station address	4	REM_IP_ADDR[1]	USint	0		5	REM_IP_ADDR[2]	USint	0		6	REM_IP_ADDR[3]	USint	0		7	REM_IP_ADDR[4]	USint	0		8	REM_PORT_NR	UInt	4.0	remote port number	9	RESERVED	Word	6.0	unused; has to be 0
Send_UDP_ADDR																																																								
Name	Datentyp	Offset	Startwert	Kommentar																																																				
1	Static																																																							
2	Addr_Data	TADDR_Param	0.0																																																					
3	REM_IP_ADDR	Array[1..4] of USint	0.0	remote station address																																																				
4	REM_IP_ADDR[1]	USint	0																																																					
5	REM_IP_ADDR[2]	USint	0																																																					
6	REM_IP_ADDR[3]	USint	0																																																					
7	REM_IP_ADDR[4]	USint	0																																																					
8	REM_PORT_NR	UInt	4.0	remote port number																																																				
9	RESERVED	Word	6.0	unused; has to be 0																																																				

### 10.2.2.6 T\_CONFIG

Die Anweisung T\_CONFIG ändert die IP-Konfigurationsparameter des PROFINET-Ports über das Anwenderprogramm. Dadurch ist die dauerhafte Änderung bzw. Einstellung der folgenden Funktionen möglich:

- Stationsname
- IP-Adresse
- Subnetzmaske
- Router-Adresse

#### Hinweis

Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU können Sie über das Optionsfeld "IP-Adresse auf anderem Weg einstellen" (Seite 484) die IP-Adresse online oder über die Anweisung "T\_CONFIG" ändern, nachdem Sie das Programm in die CPU geladen haben. Diese Zuweisungsart der IP-Adresse gilt nur für die CPU.

Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU können Sie über das Optionsfeld "PROFINET-Gerätename auf anderem Weg einstellen" (Seite 485) den PROFINET-Gerätenamen online oder über die Anweisung "T\_CONFIG" ändern, nachdem Sie das Programm in die CPU geladen haben. Diese Zuweisungsart des PROFINET-Gerätenamens gilt nur für die CPU.

#### WARNUNG

Nachdem Sie mit T\_CONFIG einen Parameter der IP-Konfiguration geändert haben, startet die CPU neu. Die CPU geht in den Betriebszustand STOP, führt einen Warmstart durch und kehrt in den Betriebszustand RUN zurück.

Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch unkontrolliertes Betriebsverhalten der gesteuerten Geräte verursachen. Derartiges unerwartetes Betriebsverhalten des Automatisierungssystems kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Stellen Sie sicher, dass Ihr Prozess in einen sicheren Zustand übergeht, wenn die CPU als Ergebnis der Ausführung der Anweisung T\_CONFIG neu startet.

Tabelle 10- 21 Anweisung T\_CONFIG

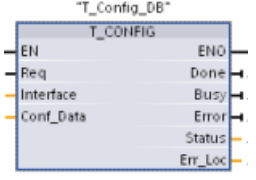
KOP/FUP	SCL	Beschreibung
	<pre>"T_CONFIG_DB" (     req:=_bool_in_,     interface:=_word_in_,     conf_Data:=_variant_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_dword_out_,     err_loc=&gt;_word_out_);</pre>	<p>Die Anweisung T_CONFIG ändert die Parameter der IP-Konfiguration über das Anwenderprogramm. T_CONFIG arbeitet asynchron. Die Ausführung erstreckt sich über mehrere Aufrufe.</p>

Tabelle 10- 22 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	Input	Bool
INTERFACE	Input	HW_Interface
CONF_DATA	Input	Variant
DONE	Output	Bool
BUSY	Output	Bool
ERROR	Output	Bool
STATUS	Output	DWord
ERR_LOC	Output	DWord

Startet die Anweisung an der steigenden Flanke.  
 ID der Netzwerkschnittstelle  
 Verweis auf die Struktur der Konfigurationsdaten; CONF\_DATA ist über einen Systemdatentyp (SDT) festgelegt.  
 • 0: Auftrag wurde noch nicht gestartet oder läuft noch.  
 • 1: Auftrag wurde fehlerfrei ausgeführt.  
 • 0: Auftrag ist beendet.  
 • 1: Der Auftrag ist noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.  
 Statusparameter mit den folgenden Werten:  
 • 0: Kein Fehler  
 • 1: Ein Fehler ist während der Bearbeitung aufgetreten. STATUS enthält ausführliche Informationen über die Fehlerart.  
 Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungs-codes für Error und Status" unten.)  
 Fehlerstelle (Feld ID und Unterfeld ID des Fehlerparameters)

Die IP-Konfigurationsdaten werden im Datenbaustein CONF\_DATA abgelegt, zusammen mit einem Variant-Pointer auf den oben angegebenen Parameter CONF\_DATA. Die erfolgreiche Ausführung der Anweisung T\_CONFIG endet mit der Übergabe der IP-Konfigurationsdaten an die Netzwerkschnittstelle. Fehler werden dem Ausgangsparameter STATUS zugewiesen.

Tabelle 10- 23 Bedingungs-codes für ERROR und STATUS

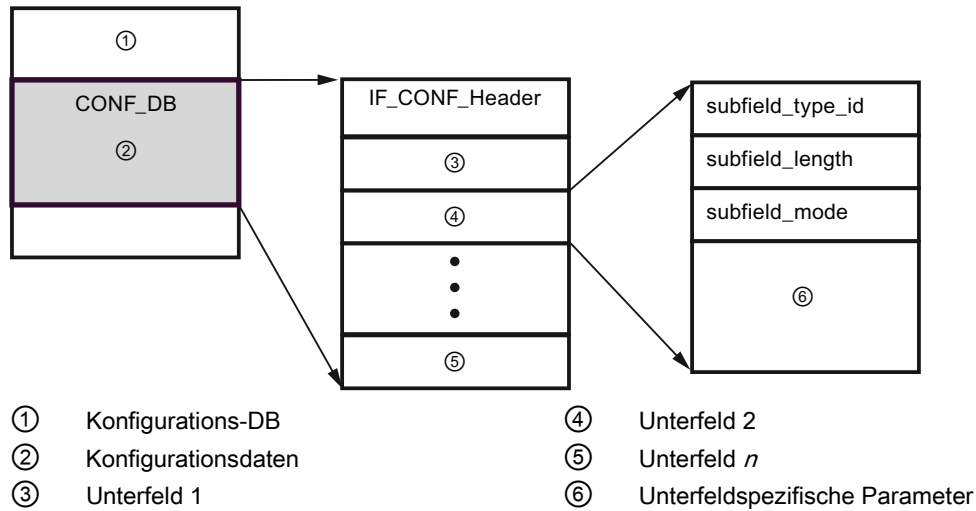
ERROR	STATUS	Beschreibung
0	00000000	Kein Fehler
0	00700000	Der Auftrag ist noch nicht beendet (BUSY = 1).



ERROR	STATUS	Beschreibung
0	00700100	Start der Auftragsausführung
0	00700200	Zwischenzeitlicher Aufruf (REQ irrelevant)
1	C08xyy00	Allgemeiner Fehler
1	C0808000	Die Parameter LADDR zur Identifikation der Schnittstelle sind ungültig.
1	C0808100	Den Parametern LADDR zur Identifikation der Schnittstelle wurde eine nicht unterstützte Hardware-Schnittstelle zugewiesen.
1	C0808200	Parameterfehler CONF_DATA: Datentyp des Variant-Pointers entspricht nicht dem Datentyp Byte.
1	C0808300	Parameterfehler CONF_DATA: Der Bereichs-Pointer ist nicht im DB des Variant-Pointers.
1	C0808400	Parameterfehler CONF_DATA: Der Variant-Pointer hat die falsche Länge.
1	C0808600	Reserviert
1	C0808700	Inkonsistenz in der Länge des CONF_DATA-Datenbausteins zur IP-Konfiguration
1	C0808800	Die Parameter der field_type_id des CONF_DATA-Datenbausteins sind ungültig. (Nur field_type_id = 0 ist zulässig.)
1	C0808900	Die Parameter der field_type_id des CONF_DATA-Datenbausteins sind ungültig oder wurden mehrmals verwendet.
1	C0808A00	Länge LEN der Parameter der IP-Konfiguration oder subfield_cnt fehlerhaft
1	C0808B00	Der Parameter ID der IP-Konfiguration ist ungültig oder wird nicht unterstützt.
1	C0808C00	Der Teilbaustein der IP-Konfiguration ist nicht korrekt abgelegt (Teilbaustein falsch, falsche Reihenfolge oder mehrmals verwendet).
1	C0808D00	Die Länge einer Anweisung LEN eines Teilbausteins ist ungültig.
1	C0808E00	Der Wert des Parameters im Teilbausteinmodus ist ungültig.
1	C0808F00	Teilbausteinkonflikt zwischen der IP-Konfiguration und einem vorherigen Teilbaustein.
1	C0809000	Die Parameter des Unterfelds sind schreibgeschützt (Beispiel: die Parameter werden von der Konfiguration angegeben oder der PNIO-Modus ist aktiviert).
1	C0809100	Reserviert
1	C0809400	Ein Parameter in der IP-Konfiguration des Teilbausteins wurde nicht definiert oder kann nicht verwendet werden.
1	C0809500	Es besteht eine Inkonsistenz zwischen einem Parameter der IP-Konfiguration des Teilbausteins und anderen Parametern.
1	C080C200	Die Anweisung kann nicht ausgeführt werden. Dieser Fehler kann auftreten, wenn beispielsweise die Kommunikation mit der Schnittstelle verloren gegangen ist.
1	C080C300	Es sind nicht genügend Ressourcen vorhanden. Dieser Fehler kann auftreten, wenn beispielsweise die Anweisung mehrmals mit verschiedenen Parametern aufgerufen wird.
1	C080C400	Kommunikation gestört. Der Fehler kann temporär auftreten und erfordert eine Wiederholung des Anwenderprogramms.
1	C080D200	Die Ausführung der Anweisung wird von der PROFINET-Schnittstelle nicht unterstützt.

### Datenbaustein CONF\_DATA

Die folgende Abbildung zeigt, wie die zu übertragenden Konfigurationsdaten im Konfigurations-DB gespeichert werden.



Die Konfigurationsdaten von CONF\_DB bestehen aus einem Feld mit einem Header (IF\_CONF\_Header) und mehreren Unterfeldern. IF\_CONF\_Header verfügt über die folgenden Elemente:

- field\_type\_id (Datentyp UInt): Null
- field\_id (Datentyp UInt): Nullabgleich
- subfield\_cnt (Datentyp UInt): Anzahl der Unterfelder

Jedes Unterfeld besteht aus einem Header (subfield\_type\_id, subfield\_length, subfield\_mode) und den unterfeldspezifischen Parametern. Jedes Unterfeld muss eine gerade Anzahl von Bytes umfassen. Der Parameter subfield\_mode unterstützt den Wert 1.

#### Hinweis

Derzeit ist nur ein Feld (IF\_CONF\_Header) zulässig. Dessen Parameter field\_type\_id und field\_id müssen den Wert 0 haben. Andere Felder mit anderen Werten für field\_type\_id und field\_id unterliegen künftigen Erweiterungen.

Im Feld IF\_CONF\_Header sind derzeit nur zwei Unterfelder "addr" (IP-Adresse) und "nos" (Name of station) zulässig.

Tabelle 10- 24 Unterstützte Unterfelder

subfield_type_id	Datentyp	Bedeutung
30	IF_CONF_V4	IP-Parameter: IP-Adresse, Subnetzmaske, Routeradresse
40	IF_CONF_NOS	PROFINET IO-Gerätename (Name of station)

Tabelle 10- 25 Elemente des Datentyps IF\_CONF\_V4

Name	Datentyp	Startwert	Beschreibung		
Id	UInt	30	subfield_type_id		
len	UInt	18	subfield_length		
mode	UInt	1	subfield_mode (1: dauerhaft)		
InterfaceAddress	IP_V4	-	Schnittstellenadresse		
ADDR	Array [1..4] of Byte				
ADDR[1]	Byte			b#16#C8	High Byte der IP-Adresse: 200
ADDR[2]	Byte			b#16#0C	High Byte der IP-Adresse: 12
ADDR[3]	Byte			b#16#01	Low Byte der IP-Adresse: 1
ADDR[4]	Byte	b#16#90	Low Byte der IP-Adresse: 144		
SubnetMask	IP_V4	-	Subnetzmaske		
ADDR	Array [1..4] of Byte				
ADDR[1]	Byte			b#16#FF	High Byte der Subnetzmaske: 255
ADDR[2]	Byte			b#16#FF	High Byte der Subnetzmaske: 255
ADDR[3]	Byte			b#16#FF	Low Byte der Subnetzmaske: 255
ADDR[4]	Byte	b#16#00	Low Byte der Subnetzmaske: 0		
DefaultRouter	IP_V4	-	Standard-Router		
ADDR	Array [1..4] of Byte				
ADDR[1]	Byte			b#16#C8	High Byte des Routers: 200
ADDR[2]	Byte			b#16#0C	High Byte des Routers: 12
ADDR[3]	Byte			b#16#01	Low Byte des Routers: 1
ADDR[4]	Byte	b#16#01	Low Byte des Routers: 1		

Tabelle 10- 26 Elemente des Datentyps IF\_CONF\_NOS

Name	Datentyp	Startwert	Beschreibung
id	UInt	40	subfield_type_id
len	UInt	246	subfield_length
mode	UInt	1	subfield_mode (1: dauerhaft)
Nos (Name of station)	Array[1..240] of Byte	0	Stationsname: Sie müssen das ARRAY ab dem ersten Byte belegen. Wenn das ARRAY länger als der zuzuweisende Stationsname ist, müssen Sie nach dem eigentlichen Stationsnamen ein Null-Byte eingeben (in Übereinstimmung mit IEC 61158-6-10). Andernfalls wird nos zurückgewiesen und die Anweisung "T_CONFIG und TC_CONFIG (Seite 479)" gibt den Fehlercode DW#16#C0809400 in STATUS ein. Wenn Sie das erste Byte mit 0 belegen, wird der Stationsname gelöscht.

Der Stationsname unterliegt den folgenden Einschränkungen:

- Eine Namenskomponente innerhalb des Stationsnamens, d.h. eine Zeichenfolge zwischen zwei Punkten, darf nicht länger sein als 63 Zeichen.
- Sonderzeichen wie Umlaute, Klammern, Unterstriche, Schrägstriche, Leerzeichen usw. sind nicht zulässig. Das einzig zulässig Sonderzeichen ist der Bindestrich.

- Der Stationsname darf nicht mit dem Zeichen "-" beginnen oder enden.
- Der Stationsname darf nicht mit einer Zahl beginnen.
- Das Format n.n.n.n (n = 0 ... 999) ist für den Stationsnamen nicht zulässig.
- Der Stationsname darf nicht mit der Zeichenfolge "Port-xyz" oder "Port-xyz-abcde" (a, b, c, d, e, x, y, z = 0 ... 9) beginnen.

**Hinweis**

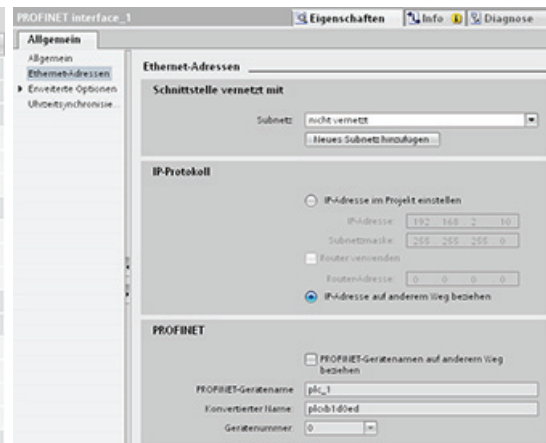
Sie können auch ein ARRAY "nos" erstellen, das kürzer ist als 240 Byte. Die Mindestlänge ist jedoch 2 Byte. Dann müssen Sie die Variable "len" (Länge des Unterfelds) entsprechend anpassen.

**IP-Parameter ändern**

Im folgenden Beispiel werden im Unterfeld "addr" die Parameter "InterfaceAddress" (IP-Adresse), "SubnetMask" und "DefaultRouter" (IP-Router) geändert. Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU muss das Optionsfeld "IP-Adresse auf anderem Weg einstellen" aktiviert sein, um die IP-Adresse über die Anweisung "T\_CONFIG" ändern zu können, nachdem Sie das Programm in die CPU geladen haben.

Tabelle 10- 27 IP-Parameter ändern

CONF_DATA_1			
	Name	Datentyp	Startwert
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	1
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	ADDR[1]	Byte	192
14	ADDR[2]	Byte	168
15	ADDR[3]	Byte	2
16	ADDR[4]	Byte	30
17	SubnetMask	IP_V4	
18	ADDR	array [1..4] of Byte	
19	ADDR[1]	Byte	255
20	ADDR[2]	Byte	255
21	ADDR[3]	Byte	255
22	ADDR[4]	Byte	0
23	DefaultRouter	IP_V4	
24	ADDR	array [1..4] of Byte	
25	ADDR[1]	Byte	192
26	ADDR[2]	Byte	168
27	ADDR[3]	Byte	2
28	ADDR[4]	Byte	1

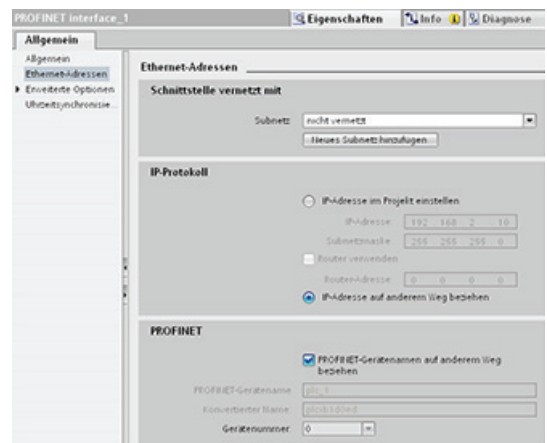


## IP-Parameter und PROFINET IO-Gerätenamen ändern

Im folgenden Beispiel werden beide Unterfelder "addr" und "nos" (Name of station) geändert. Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU muss das Optionsfeld "PROFINET-Gerätename auf anderem Weg einstellen" aktiviert sein, um den PROFINET-Gerätenamen über die Anweisung "T\_CONFIG" ändern zu können, nachdem Sie das Programm in die CPU geladen haben.

Tabelle 10- 28 IP-Parameter und PROFINET IO-Gerätenamen ändern

CONF_DATA_2			
	Name	Datentyp	Startwert
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	Uint	0
5	FieldId	Uint	0
6	SubfieldCount	Uint	2
7	addr	IF_CONF_v4	
8	Id	Uint	30
9	Length	Uint	18
10	Mode	Uint	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	SubnetMask	IP_V4	
14	ADDR	array [1..4] of Byte	
15	DefaultRouter	IP_V4	
16	ADDR	array [1..4] of Byte	
17	nos	IF_CONF_IPOS	
18	Id	Uint	40
19	Length	Uint	246
20	Mode	Uint	1
21	IPOS	array [1..240] of Byte	



### 10.2.2.7 Gemeinsame Parameter für Anweisungen

#### Eingangsparameter REQ

Viele der Anweisungen für die offene Benutzerkommunikation haben einen Eingang REQ, der die Anweisung bei einer steigenden Flanke (0 nach 1) initiiert. Der Eingang REQ muss während der Ausführung einer Anweisung 1 (WAHR) sein, doch der Eingang REQ kann so lange wie gewünscht WAHR bleiben. Die Anweisung stößt erst dann eine andere Anweisung an, wenn sie mit Eingang REQ = FALSCH ausgeführt wird, so dass die Anweisung den Zustand des Eingangs REQ zurücksetzen kann. Dies ist erforderlich, damit die Anweisung die steigende Flanke zum Starten der nächsten Anweisung erkennen kann.

Wenn Sie eine dieser Anweisungen in Ihr Programm einfügen, werden Sie von STEP 7 aufgefordert, den Instanz-DB anzugeben. Verwenden Sie für jeden Anweisungsauftrag einen eindeutigen DB. Dadurch wird sichergestellt, dass jede Anweisung die Eingänge wie REQ ordnungsgemäß verarbeitet.

**Eingangsparameter ID**

Dies ist ein Verweis auf die "Lokale ID (hex)" in der "Netzansicht" unter "Geräte & Netze" in STEP 7 und es ist die ID des Netzwerks, das Sie für diesen Kommunikationsbaustein verwenden möchten. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen.

**Ausgangsparameter DONE, NDR, ERROR und STATUS**

Diese Anweisungen haben Ausgänge, die den Ausführungszustand anzeigen:

Tabelle 10- 29   Ausgabeparameter der Anweisungen für die offene Benutzerkommunikation

Parameter	Datentyp	Standard	Beschreibung
DONE	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehler abgeschlossen wurde; andernfalls FALSCH.
NDR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die angeforderte Aktion fehlerfrei abgeschlossen und neue Daten empfangen wurden; andernfalls FALSCH.
BUSY	Bool	FALSCH	Wird, wenn aktiv, auf WAHR gesetzt, um Folgendes anzuzeigen: <ul style="list-style-type: none"> <li>• Der Auftrag ist noch nicht beendet.</li> <li>• Es kann kein neuer Auftrag gestartet werden.</li> </ul> Wird auf FALSCH gesetzt, wenn der Auftrag beendet ist.
ERROR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehlern abgeschlossen wurde, der entsprechende Fehlercode befindet sich in STATUS; andernfalls FALSCH.
STATUS	Word	0	Ergebniszustand: <ul style="list-style-type: none"> <li>• Wird das Bit DONE oder NDR gesetzt, wird STATUS auf 0 oder auf einen Informationscode gesetzt.</li> <li>• Wird das Bit ERROR gesetzt, wird STATUS auf einen Fehlercode gesetzt.</li> <li>• Wird keines der oben aufgeführten Bits gesetzt, kann die Anweisung Statusergebnisse zurückgeben, die den aktuellen Zustand der Funktion beschreiben.</li> </ul> Der Wert in STATUS wird während der Ausführung der Funktion gehalten.

**Hinweis**

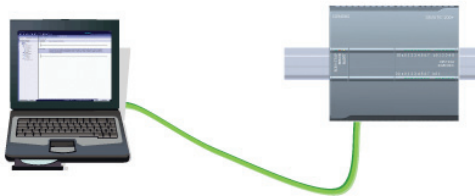
Beachten Sie, dass DONE, NDR und ERROR nur eine Ausführung lang gesetzt sind.

### Eingeschränkte TSAPs und Portnummern für passive ISO- und TCP-Kommunikation

Wenn Sie mit der Operation TCON eine passive Kommunikationsverbindung einrichten und aufbauen, dürfen die folgenden Portadressen nicht verwendet werden:

- ISO TSAP (passiv):
  - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
  - 10.00, 10.01, 11.00, 11.01, ... BF0,00, BF0,01
- TCP-Port (passiv): 5001, 102, 123, 20, 21, 25, 34962, 34963, 34964, 80
- UDP-Port (passiv): 161, 34962, 34963, 34964

### 10.2.3 Kommunikation mit einem Programmiergerät



Eine CPU kann mit einem Programmiergerät STEP 7 in einem Netzwerk kommunizieren.

Berücksichtigen Sie beim Einrichten der Kommunikation zwischen einer CPU und einem Programmiergerät Folgendes:

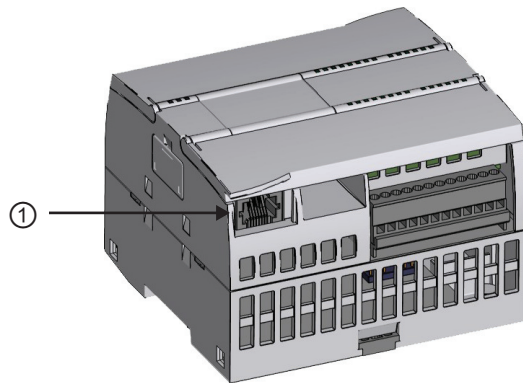
- Konfiguration/Setup: Eine Hardwarekonfiguration ist erforderlich.
- Für die Kommunikation zwischen zwei Teilnehmern ist kein Ethernet-Switch erforderlich; erst wenn mehr als zwei Geräte in einem Netzwerk vorhanden sind, wird ein Ethernet-Switch benötigt.

#### 10.2.3.1 Hardware-Kommunikationsverbindung herstellen

Die PROFINET-Schnittstelle stellt die physische Verbindung zwischen einem Programmiergerät und einer CPU her. Da in der CPU die Auto-Cross-Over-Funktionalität implementiert ist, kann für die Schnittstelle ein Standard- oder Crossover-Ethernet-Kabel verwendet werden. Ein Ethernet-Switch wird für die direkte Verbindung des Programmiergeräts mit der CPU nicht benötigt.

Zum Herstellen der Hardwareverbindung zwischen einem Programmiergerät und einer CPU gehen Sie folgendermaßen vor:

1. Installieren Sie die CPU (Seite 52).
2. Schließen Sie das Ethernet-Kabel wie unten gezeigt am PROFINET-Anschluss an.
3. Schließen Sie das Ethernet-Kabel an das Programmiergerät an.



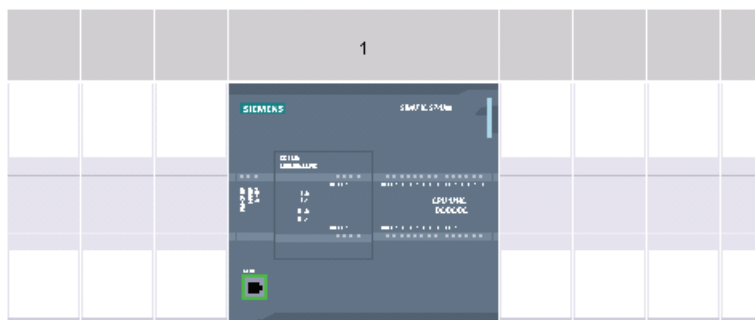
① PROFINET-Anschluss

Für die PROFINET-Verbindung ist eine optionale Zugentlastung verfügbar.

### 10.2.3.2 Konfigurieren der Geräte

Wenn Sie bereits ein Projekt mit einer CPU angelegt haben, öffnen Sie Ihr Projekt in STEP 7.

Sonst legen Sie ein Projekt an und fügen eine CPU (Seite 130) in den Baugruppenträger ein. Im folgenden Projekt sehen Sie eine CPU in der "Gerätesicht".





### 10.2.3.3 IP-Adressen zuweisen

#### IP-Adressen zuweisen

In einem PROFINET-Netzwerk benötigt jedes Gerät außerdem eine IP-Adresse (Internet-Protocol-Adresse). Mit Hilfe dieser Adresse kann ein Gerät Daten über ein komplexeres Netzwerk liefern:

- Verwenden Sie Programmiergeräte oder andere Netzwerkgeräte mit integrierter Adapterkarte, die an Ihr Werks-LAN angeschlossen ist, oder mit Ethernet/USB-Adapterkarte, die an ein getrenntes Netzwerk angeschlossen ist, so müssen Sie diesen Geräten IP-Adressen zuweisen. Weitere Informationen finden Sie unter "IP-Adressen zu Programmier- und Netzwerkgeräten zuweisen" (Seite 143).
- Sie können einer CPU oder einem Netzwerkgerät auch online eine IP-Adresse zuweisen. Dies ist besonders bei der ersten Gerätekonfiguration nützlich. Weitere Informationen finden Sie unter "IP-Adresse zu einer CPU online zuweisen (Seite 143)".
- Nachdem Sie Ihre CPU oder Ihr Netzwerkgerät in Ihrem Projekt konfiguriert haben, können Sie die Parameter für die PROFINET-Schnittstelle konfigurieren, um die IP-Adresse aufzunehmen. Weitere Informationen finden Sie unter "IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 145).

### 10.2.3.4 Testen Ihres PROFINET-Netzwerks

Nach Abschluss der Konfiguration müssen Sie Ihr Projekt in die CPU laden. Alle IP-Adressen werden beim Laden des Projekts konfiguriert.

Die Funktion "Laden in Gerät" der CPU und der zugehörigen Dialog "Erweitertes Laden" kann alle erreichbaren Netzwerkgeräte anzeigen und zusätzlich angeben, ob allen Geräten eindeutige IP-Adressen zugeordnet wurden. Ausführliche Informationen finden Sie unter "Testen des PROFINET-Netzwerks" (Seite 150).

## 10.2.4 Kommunikation HMI/PLC



Die CPU unterstützt PROFINET-Kommunikationsverbindungen mit HMIs. Zum Einrichten der Kommunikation zwischen CPUs und HMIs müssen die folgenden Bedingungen erfüllt sein:

Konfiguration/Setup:

- Der PROFINET-Port der CPU muss für die Verbindung mit der HMI konfiguriert sein.
- Die HMI muss eingerichtet und konfiguriert sein.

- Die HMI-Konfigurationsdaten sind Teil des CPU-Projekts und können im Projekt konfiguriert und geladen werden.
- Für die Kommunikation zwischen zwei Teilnehmern ist kein Ethernet-Switch erforderlich; erst wenn mehr als zwei Geräte in einem Netzwerk vorhanden sind, wird ein Ethernet-Switch benötigt.

---

**Hinweis**

Der auf dem Baugruppenträger montierte Ethernet-Switch CSM1277 mit 4 Ports dient zum Anschließen Ihrer CPUs und HMI-Geräte. Der PROFINET-Anschluss an der CPU enthält keinen Ethernet-Switch.

---

Unterstützte Funktionen:

- Die HMI kann Daten der CPU lesen/schreiben.
- Meldungen können anhand von Informationen der CPU ausgelöst werden.
- Systemdiagnose

Tabelle 10- 30 Erforderliche Schritte zum Konfigurieren der Kommunikation zwischen einer HMI und einer CPU

Schritt	Aufgabenstellung
1	Hardware-Kommunikationsverbindung herstellen Eine PROFINET-Schnittstelle stellt die physische Verbindung zwischen einer HMI und einer CPU her. Da in der CPU die Auto-Cross-Over-Funktionalität implementiert ist, kann für die Schnittstelle ein Standard- oder Crossover-Ethernet-Kabel verwendet werden. Ein Ethernet-Switch wird für die direkte Verbindung der HMI mit der CPU nicht benötigt. Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Hardware-Kommunikationsverbindung herstellen (Seite 487)".
2	Geräte konfigurieren Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Geräte konfigurieren (Seite 488)".
3	Konfigurieren der logischen Netzwerkverbindungen zwischen einer HMI und einer CPU Weitere Informationen finden Sie unter "Kommunikation zwischen HMI und PLC: Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten" (Seite 491).
4	IP-Adresse in Ihrem Projekt konfigurieren Gehen Sie auf dieselbe Weise vor. Sie müssen jedoch die IP-Adressen für die HMI und die CPU konfigurieren. Weitere Informationen finden Sie unter "Gerätekonfiguration: IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 147).
5	Testen des PROFINET-Netzwerks Sie müssen die Konfiguration für jede CPU und jedes HMI-Gerät laden. Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Testen des PROFINET-Netzwerks" (Seite 150).

### 10.2.4.1 Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten

Nachdem Sie den Baugruppenträger mit der CPU konfiguriert haben, können Sie Ihre Netzwerkverbindungen einrichten.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Klicken Sie zunächst auf das Register "Verbindungen" und wählen Sie dann über die Klappliste rechts den Verbindungstyp aus (z. B. eine ISO-on-TCP-Verbindung).

Um eine PROFINET-Verbindung zu erstellen, klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 137).

### 10.2.5 Kommunikation PLC/PLC



Eine CPU kann mit den Anweisungen TSEND\_C und TRCV\_C mit einer anderen CPU im Netzwerk kommunizieren.

Berücksichtigen Sie beim Einrichten der Kommunikation zwischen zwei CPUs Folgendes:

- Konfiguration/Setup: Eine Hardwarekonfiguration ist erforderlich.
- Unterstützte Funktionen: Lesen/Schreiben von Daten in eine vernetzte CPU
- Für die Kommunikation zwischen zwei Teilnehmern ist kein Ethernet-Switch erforderlich; erst wenn mehr als zwei Geräte in einem Netzwerk vorhanden sind, wird ein Ethernet-Switch benötigt.

Tabelle 10- 31 Erforderliche Schritte zum Konfigurieren der Kommunikation zwischen zwei CPUs

Schritt	Aufgabenstellung
1	Hardware-Kommunikationsverbindung herstellen Eine PROFINET-Schnittstelle stellt die physische Verbindung zwischen zwei CPUs her. Da in der CPU die Auto-Cross-Over-Funktionalität implementiert ist, kann für die Schnittstelle ein Standard- oder Crossover-Ethernet-Kabel verwendet werden. Ein Ethernet-Switch wird für die Verbindung der zwei CPUs nicht benötigt. Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Hardware-Kommunikationsverbindung herstellen (Seite 487)".
2	Geräte konfigurieren Sie müssen zwei CPUs in Ihrem Projekt konfigurieren. Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Geräte konfigurieren (Seite 488)".

Schritt	Aufgabenstellung
3	Konfigurieren der logischen Netzwerkverbindungen zwischen zwei CPUs Weitere Informationen finden Sie unter "Kommunikation zwischen PLC und PLC: Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten" (Seite 492).
4	IP-Adresse in Ihrem Projekt konfigurieren Gehen Sie auf dieselbe Weise vor. Sie müssen jedoch die IP-Adressen für zwei CPUs konfigurieren (z.B. PLC_1 und PLC_2). Weitere Informationen finden Sie unter "Gerätekonfiguration: IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 147).
5	Sende- und Empfangsparameter konfigurieren Sie müssen die Anweisungen TSEND_C und TRCV_C in beiden CPUs konfigurieren, um die Kommunikation zwischen ihnen zu ermöglichen. Weitere Informationen finden Sie unter "Einrichten der Kommunikation zwischen zwei CPUs: Send- und Empfangsparameter konfigurieren (Seite 493)".
6	Testen des PROFINET-Netzwerks Sie müssen die Konfiguration für jede CPU laden. Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Testen des PROFINET-Netzwerks" (Seite 150).

### 10.2.5.1 Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten

Nachdem Sie den Baugruppenträger mit der CPU konfiguriert haben, können Sie Ihre Netzwerkverbindungen einrichten.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Klicken Sie zunächst auf das Register "Verbindungen" und wählen Sie dann über die Klappliste rechts den Verbindungstyp aus (z. B. eine ISO-on-TCP-Verbindung).

Um eine PROFINET-Verbindung zu erstellen, klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 137).

### 10.2.5.2 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

#### Allgemeine Parameter konfigurieren

Sie legen die Kommunikationsparameter im Dialog "Eigenschaften" der Kommunikationsanweisung fest. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung selektiert haben.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 138)".

Die zu verwendenden TSAPs oder Ports definieren Sie im Dialog "Verbindungsparameter" unter "Adressdetails". Der TSAP oder Port einer Verbindung in der CPU wird im Feld "Lokaler TSAP" eingegeben. Der TSAP oder Port für die Verbindung in der Partner-CPU wird im Feld "Partner-TSAP" eingegeben.

### 10.2.5.3 Sende- und Empfangsparameter konfigurieren

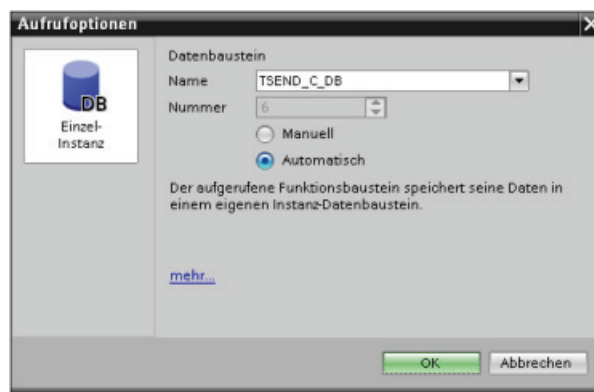
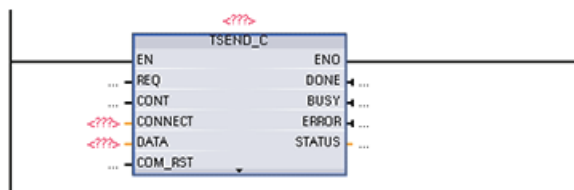
Über Kommunikationsbausteine (z. B. TSEND\_C und TRCV\_C) werden Verbindungen zwischen zwei CPUs hergestellt. Damit die PROFINET-Kommunikation von den CPUs aufgenommen werden kann, müssen die Parameter zum Senden und Empfangen von Meldungen eingerichtet werden. Diese Parameter legen fest, wie die Kommunikation abläuft, wenn Meldungen von einem Zielgerät gesendet oder von diesem empfangen werden.

#### Sendeparameter für TSEND\_C konfigurieren

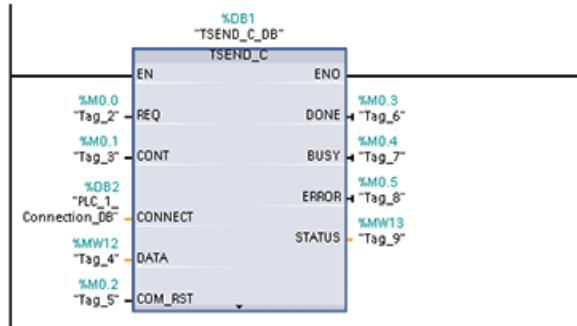
#### Anweisung TSEND\_C

Die Anweisung TSEND\_C (Seite 459) baut eine Kommunikationsverbindung zu einer Partnerstation auf. Die Verbindung wird eingerichtet, aufgebaut und automatisch überwacht, bis eine Anweisung die Verbindung trennt. Die Anweisung TSEND\_C verbindet die Funktionen von TCON, TDISCON und TSEND.

In STEP 7 können Sie in der Gerätekonfiguration einstellen, wie Daten mit einer Anweisung TSEND\_C gesendet werden sollen. Zuerst fügen Sie die Anweisung aus dem Ordner "Kommunikation" in der Taskcard "Anweisungen" in das Programm ein. Die Anweisung TSEND\_C wird zusammen mit dem Dialog "Aufrufoptionen" angezeigt, in dem Sie einen DB zum Speichern der Parameter der Anweisung zuweisen können.



Sie können den Ein- und Ausgängen Speicherplätze im Variablenspeicher zuweisen. Dies wird in der folgenden Abbildung gezeigt:



### Allgemeine Parameter konfigurieren

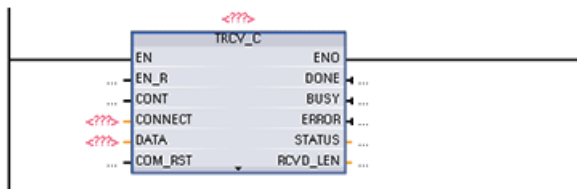
Sie können die Parameter im Dialog "Eigenschaften" der Anweisung TSEND\_C festlegen. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung TSEND\_C selektiert haben.

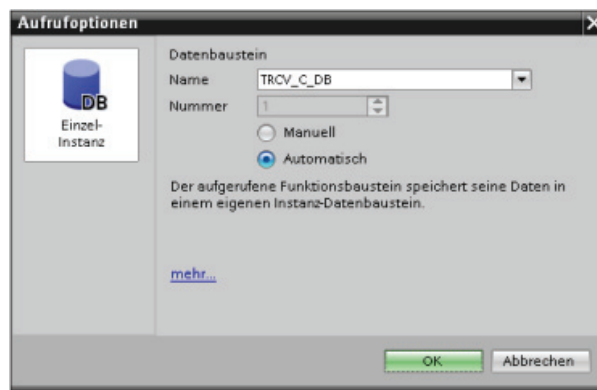
### Empfangsparameter für TRCV\_C konfigurieren

#### Anweisung TRCV\_C

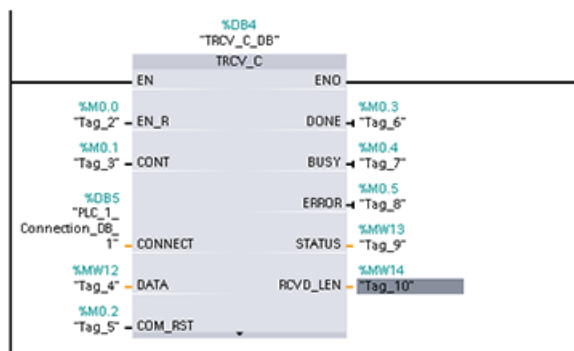
Die Anweisung TRCV\_C (Seite 459) baut eine Kommunikationsverbindung zu einer Partnerstation auf. Die Verbindung wird eingerichtet, aufgebaut und automatisch überwacht, bis eine Anweisung die Verbindung trennt. Die Anweisung TRCV\_C verbindet die Funktionen von TCON, TDISCON und TRCV.

In STEP 7 können Sie in der CPU-Konfiguration einstellen, wie eine Anweisung TRCV\_C Daten empfangen soll. Zuerst fügen Sie die Anweisung aus dem Ordner "Kommunikation" in der Taskcard "Anweisungen" in das Programm ein. Die Anweisung TRCV\_C wird zusammen mit dem Dialog "Aufrufoptionen" angezeigt, in dem Sie einen DB zum Speichern der Parameter der Anweisung zuweisen können.





Sie können den Ein- und Ausgängen Speicherplätze im Variablenspeicher zuweisen. Dies wird in der folgenden Abbildung gezeigt:



### Allgemeine Parameter konfigurieren

Sie können die Parameter im Dialog "Eigenschaften" der Anweisung TRCV\_C festlegen. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung TRCV\_C selektiert haben.

## 10.2.6 CPU und PROFINET IO-Gerät konfigurieren

### PROFINET IO-Gerät hinzufügen

Verwenden Sie im Portal "Geräte & Netze" den Hardwarekatalog, um PROFINET IO-Geräte hinzuzufügen.

---



#### Hinweis

Um ein PROFINET IO-Gerät hinzuzufügen, können Sie STEP 7 Professional oder Basic ab V11 verwenden.

---

Zum Einfügen eines ET200S IO-Geräts erweitern Sie zum Beispiel die folgenden Behälter im Hardwarekatalog: Dezentrale Peripherie, ET200S, Schnittstellenmodule und PROFINET. Sie können dann das Schnittstellenmodul in der Liste der ET200S-Geräte (die nach der Bestellnummer sortiert sind) auswählen und das ET200S IO-Gerät hinzufügen.

Tabelle 10- 32 ET200S IO-Gerät zur Gerätekonfiguration hinzufügen

IO-Gerät einfügen	Ergebnis
	

Sie können nun das PROFINET IO-Gerät mit der CPU verbinden:

1. Klicken Sie auf dem Gerät mit der rechten Maustaste auf den Link "Nicht zugewiesen" und wählen Sie im Kontextmenü den Befehl "Neue IO-Steuerung zuweisen" aus, um den Dialog "IO-Steuerung auswählen" aufzurufen.
2. Wählen Sie in der Liste der IO-Steuerungen im Projekt Ihre S7-1200 CPU aus (in diesem Beispiel "PLC\_1").
3. Klicken Sie auf "OK", um die Netzwerkverbindung anzulegen.

### Logische Netzwerkverbindungen konfigurieren

Nachdem Sie den Baugruppenträger mit der CPU konfiguriert haben, können Sie Ihre Netzwerkverbindungen einrichten.

Im Portal "Geräte & Netze" können Sie in der "Netzsicht" die Geräte in Ihrem Projekt vernetzen. Um eine PROFINET-Verbindung zu erstellen, klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 137).

### CPUs und Gerätenamen zuweisen

Netzwerkverbindungen zwischen den Geräten weisen auch das PROFINET IO-Gerät der CPU zu, was erforderlich ist, damit die CPU das Gerät steuern kann. Um diese Zuweisung zu ändern, klicken Sie auf den PLC-Namen, der auf dem PROFINET IO-Gerät angezeigt wird. Daraufhin wird ein Dialog geöffnet, in dem Sie das PROFINET IO-Gerät von der aktuellen CPU trennen und neu zuweisen oder auch ohne Zuweisung lassen können.

Den Geräten in Ihrem PROFINET-Netzwerk muss ein Name zugewiesen sein, damit Sie eine Verbindung zur CPU herstellen können. Sie weisen Ihren PROFINET-Geräten in der "Netzsicht" Namen zu, wenn die Geräte noch keinen Namen haben oder wenn der Name eines Geräts geändert werden soll. Klicken Sie hierfür mit der rechten Maustaste auf das PROFINET IO-Gerät und wählen Sie "Gerätename zuweisen".



Den Namen eines PROFINET IO-Geräts müssen Sie sowohl im STEP 7-Projekt als auch über das Werkzeug "Online & Diagnose" im Konfigurationsspeicher des PROFINET IO-Geräts zuweisen (z. B. im Konfigurationsspeicher eines ET200 S-Schnittstellenmoduls). Fehlt ein Name oder entsprechen sich die Namen an den beiden Speicherorten nicht, funktioniert der Modus für den PROFINET IO-Datenaustausch nicht. Weitere Informationen finden Sie unter "Online- und Diagnose-Tools: Einem PROFINET-Gerät online einen Namen zuweisen (Seite 720)".

## IP-Adressen zuweisen

In einem PROFINET-Netzwerk benötigt jedes Gerät außerdem eine IP-Adresse (Internet-Protocol-Adresse). Mit Hilfe dieser Adresse kann ein Gerät Daten über ein komplexeres Netzwerk liefern:

- Verwenden Sie Programmiergeräte oder andere Netzwerkgeräte mit integrierter Adapterkarte, die an Ihr Werks-LAN angeschlossen ist, oder mit Ethernet/USB-Adapterkarte, die an ein getrenntes Netzwerk angeschlossen ist, so müssen Sie diesen Geräten IP-Adressen zuweisen. Weitere Informationen finden Sie unter "IP-Adressen zu Programmier- und Netzwerkgeräten zuweisen" (Seite 143).
- Sie können einer CPU oder einem Netzwerkgerät auch online eine IP-Adresse zuweisen. Dies ist besonders bei der ersten Gerätekonfiguration nützlich. Weitere Informationen finden Sie unter "IP-Adresse zu einer CPU online zuweisen (Seite 145)".
- Nachdem Sie Ihre CPU oder Ihr Netzwerkgerät in Ihrem Projekt konfiguriert haben, können Sie die Parameter für die PROFINET-Schnittstelle konfigurieren, um die IP-Adresse aufzunehmen. Weitere Informationen finden Sie unter "IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 147).

## IO-Zykluszeit konfigurieren

Ein PROFINET IO-Gerät wird innerhalb des Zeitraums "IO-Zyklus" von der CPU mit neuen Daten versorgt. Die Aktualisierungszeit kann für jedes Gerät separat konfiguriert werden und legt den Zeitraum fest, in dem Daten von der CPU zum und vom Gerät übertragen werden.

STEP 7 berechnet die Aktualisierungszeit für den "IO-Zyklus" automatisch in den Standardeinstellungen für jedes Gerät des PROFINET-Netzwerks und berücksichtigt dabei das Volumen der auszutauschenden Daten und die Anzahl der dieser Steuerung zugewiesenen Geräte. Wenn Sie nicht möchten, dass die Aktualisierungszeit automatisch berechnet wird, können Sie diese Einstellung ändern.

Sie legen die Parameter für den "IO-Zyklus" im Konfigurationsdialog "Eigenschaften" des PROFINET IO-Geräts fest. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung selektiert haben.

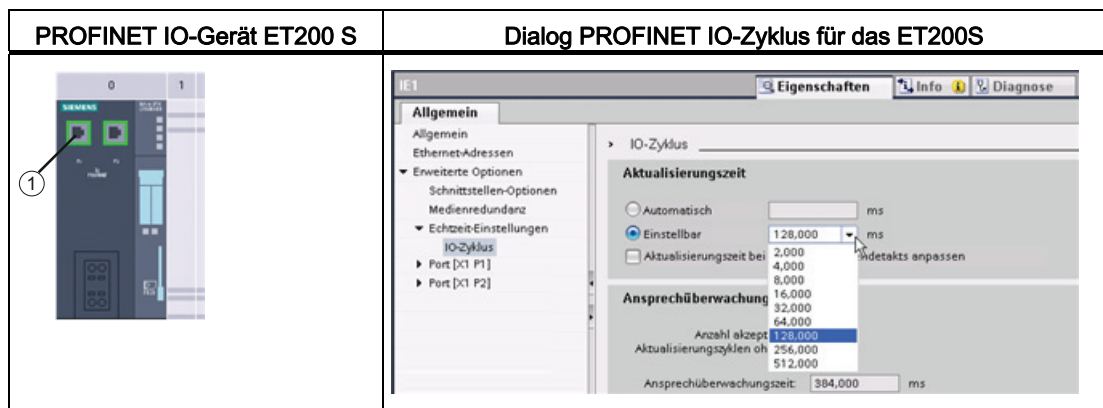
Klicken Sie in der "Gerätesicht" des PROFINET IO-Geräts auf den PROFINET-Port. Im Dialog "PROFINET-Schnittstelle" rufen Sie die Parameter "IO-Zyklus" mit der folgenden Menüauswahl auf:

- "Erweiterte Optionen"
- "Echtzeiteinstellungen"
- "IO-Zyklus"

Definieren Sie die "Aktualisierungszeit" für den IO-Zyklus wie folgt:

- Wenn eine geeignete Aktualisierungszeit automatisch berechnet werden soll, wählen Sie "Automatisch".
- Um die Aktualisierungszeit selbst einzustellen, wählen Sie "Kann eingestellt werden" und geben die gewünschte Aktualisierungszeit in ms ein.
- Um die Konsistenz zwischen dem Sendetakt und der Aktualisierungszeit sicherzustellen, aktivieren Sie die Option "Aktualisierungszeit anpassen, wenn sich der Sendetakt ändert". Diese Option gewährleistet, dass keine Aktualisierungszeit eingestellt wird, die kleiner als der Sendetakt ist.

Tabelle 10- 33 PROFINET IO-Zykluszeit für das ET200S konfigurieren



① PROFINET-Port

## 10.2.7 Diagnose

### Diagnosealarm-Organisationsbaustein (OB82)

Wenn ein Modul mit Diagnosefähigkeit, für das Sie den Diagnosealarm aktiviert haben, einen veränderten Diagnosezustand erkennt, sendet es eine Diagnosealarmanforderung an die CPU, und zwar in den folgenden Situationen:

- Dieses Modul hat ein Problem erkannt (beispielsweise einen Drahtbruch) oder eine Komponente muss gewartet werden oder beides (kommendes Ereignis).
- Das Problem wurde behoben oder besteht nicht mehr und keine weiteren Komponenten müssen gewartet werden (gehendes Ereignis).

Ist OB82 nicht vorhanden, werden diese Fehler in den Diagnosepuffer geschrieben. Die CPU ergreift keine Maßnahmen und geht auch nicht in STOP.

Ist OB82 vorhanden, kann das Betriebssystem in Reaktion auf ein kommendes Ereignis OB82 aufrufen. Sie müssen OB82 erstellen, und mit diesem OB können Sie die lokale Fehlerbearbeitung und eine detailliertere Reaktion auf kommende Ereignisse konfigurieren.

Wenn Sie eine DPV1-fähige CPU nutzen, können Sie mit Hilfe der Anweisung RALRM spezifischere Informationen zum Alarm abrufen, die über die Startinformationen von OB82 hinausgehen.

### Peripheriezugriffsalarme

Diese Fehler werden in den Diagnosepuffer geschrieben. Die CPU ergreift keine Maßnahmen und geht auch nicht in STOP. Zu den Fehlern, die in den Diagnosepuffer geschrieben werden, gehören:

- Modulfehler
- Modul stimmt nicht überein
- Modul fehlt

### IO-Zugriffsfehler

Diese Fehler werden in den Diagnosepuffer geschrieben. Die CPU ergreift keine Maßnahmen und geht auch nicht in STOP.

#### 10.2.8 Anweisungen für die dezentrale E/A

Weitere Informationen zur Verwendung der Anweisungen für die dezentrale Peripherie mit diesen Kommunikationsnetzwerken finden Sie unter "Dezentrale Peripherie (PROFINET, PROFIBUS oder AS-i)" (Seite 289).

#### 10.2.9 Diagnoseanweisungen

Weitere Informationen zur Verwendung dieser Anweisungen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseanweisungen" (Seite 315).

#### 10.2.10 Diagnoseereignisse für die dezentrale Peripherie

Weitere Informationen zur Verwendung dieser Diagnoseinformationen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseereignisse für die dezentrale Peripherie" (Seite 315).

## 10.3 PROFIBUS

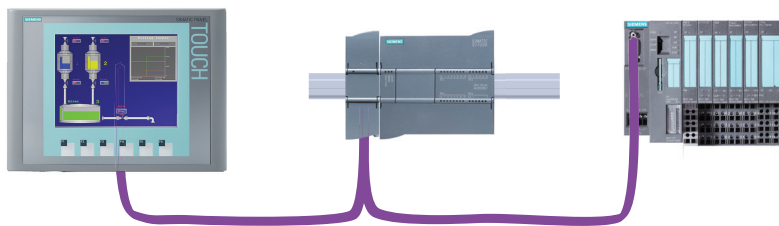
Ein PROFIBUS-System nutzt einen Bus-Master, um Slavegeräte abzufragen, die an mehreren Stellen auf einem seriellen RS485-Bus verteilt sind. Ein PROFIBUS-Slave ist ein beliebiges Peripheriegerät (E/A-Wandler, Ventil, Motorantrieb oder Messgerät), das Daten verarbeitet und die Ausgabe an den Master sendet. Der Slave stellt eine passive Station im Netzwerk dar, weil er keine Buszugriffsrechte besitzt. Er kann lediglich empfangene Meldungen quittieren oder auf Anforderung Antwortmeldungen an den Master senden. Alle PROFIBUS-Slaves haben die gleiche Priorität und die gesamte Netzwerkkommunikation stammt vom Master.

Ein PROFIBUS-Master stellt eine "aktive Station" im Netzwerk dar. PROFIBUS DP definiert zwei Klassen von Mastern. Ein Master der Klasse 1 (normalerweise eine zentrale programmierbare Steuerung (PLC) oder ein PC mit Spezialsoftware) bearbeitet die normale Kommunikation bzw. den Datenaustausch mit den ihm zugewiesenen Slaves. Ein Master der Klasse 2 (üblicherweise ein Konfigurationsgerät, z. B. ein Laptop oder eine Programmierkonsole für Inbetriebnahme, Wartung und Diagnose) ist ein Sondergerät, das hauptsächlich für die Inbetriebnahme von Slaves und zu Diagnosezwecken eingesetzt wird.

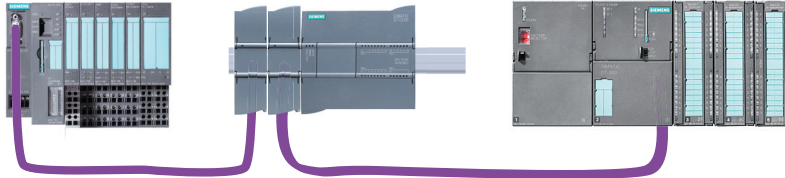
Die S7-1200 ist mit dem Kommunikationsmodul CM 1242-5 als DP-Slave an ein PROFIBUS-Netzwerk angeschlossen. Das CM 1242-5 (DP-Slave) kann der Kommunikationspartner von DP-Mastern V0/V1 sein. In der folgenden Abbildung ist die S7-1200 ein DP-Slave einer S7-300 Steuerung.



Die S7-1200 ist mit dem Kommunikationsmodul CM 1243-5 als DP-Master an ein PROFIBUS-Netzwerk angeschlossen. Das CM 1243-5 (DP-Master) kann der Kommunikationspartner von DP-Slaves V0/V1 sein. In der folgenden Abbildung ist die S7-1200 ein Master und steuert einen ET200S DP-Slave.



Wenn ein CM 1242-5 und ein CM 1243-5 zusammen installiert sind, kann eine S7-1200 gleichzeitig sowohl als Slave eines übergeordneten DP-Mastersystems als auch als Master eines untergeordneten DP-Mastersystems fungieren.



In V3.0 können Sie maximal drei PROFIBUS CMs je Station konfigurieren, wobei es sich um jede beliebige Kombination aus DP-Master- oder DP-Slave CMs handeln kann. DP-Master in einer V3.0-Implementierung können jeweils maximal 32 Slaves steuern.

In V2.2 können Sie maximal drei PROFIBUS CMs je Station konfigurieren, von denen nur eines ein DP-Master sein darf. Ein DP-Master in einer V2.2-Implementierung kann maximal 16 Slaves steuern.

## 10.3.1 Kommunikationsmodule für PROFIBUS

### 10.3.1.1 Anbindung an PROFIBUS

#### Anbindung der S7-1200 an PROFIBUS DP

Mithilfe der folgenden Kommunikationsmodule lässt sich die S7-1200 an ein PROFIBUS-Feldbusssystem anschließen:

- CM 1242-5  
Funktion als DP-Slave
- CM 1243-5  
Funktion als DP-Master Klasse 1

Bei Einbau eines CM 1242-5 und eines CM 1243-5 kann eine S7-1200 gleichzeitig folgende Funktionen ausfüllen:

- Slave eines übergeordneten DP-Mastersystems  
und
- Master eines untergeordneten DP-Mastersystems

### 10.3.1.2 Kommunikationsdienste der PROFIBUS-CMs

#### Busprotokoll

Die PROFIBUS-CMs verwenden das Protokoll PROFIBUS DP-V1.

#### PROFIBUS-Kommunikationspartner der S7-1200

Mit den beiden PROFIBUS-CMs wird der S7-1200 die Datenübertragung zu folgenden Kommunikationspartnern ermöglicht.

- CM 1242-5

Das CM 1242-5 (DP-Slave) kann Kommunikationspartner folgender DP-V0/V1-Master sein:

- SIMATIC S7-1200, S7-300, S7-400, S7-Modular Embedded Controller
- DP-Master-Baugruppen der dezentralen Peripherie SIMATIC ET200
- SIMATIC-PC-Stationen
- SIMATIC NET IE/PB Link
- Automatisierungsgeräte verschiedener Hersteller

- CM 1243-5

Das CM 1243-5 (DP-Master) kann Kommunikationspartner folgender DP-V0/V1-Slaves sein:

- Dezentralen Peripherie SIMATIC ET200
- S7-1200-CPU mit CM 1242-5
- S7-200-CPU mit PROFIBUS DP-Modul EM 277
- SINAMICS-Umrichter
- Antriebe und Aktoren verschiedener Hersteller
- Sensoren verschiedener Hersteller
- S7-300/400-CPU mit PROFIBUS-Schnittstelle
- S7-300/400-CPU mit PROFIBUS-CP (z. B. CP 342-5)
- SIMATIC PC-Stationen mit PROFIBUS-CP

## Kommunikationsarten unter DP-V1

Folgende Kommunikationsarten stehen unter DP-V1 zur Verfügung:

- Zyklische Kommunikation (CM 1242-5 und CM 1243-5)

Beide PROFIBUS-Baugruppen unterstützen zyklische Kommunikation zur Übertragung von Prozessdaten zwischen DP-Slave und DP-Master.

Die zyklische Kommunikation führt das Betriebssystem der CPU durch. Hierfür sind keine Software-Bausteine erforderlich. Die E/A-Daten werden direkt in das Prozessabbild der CPU gelesen bzw. geschrieben.

- Azyklische Kommunikation (nur CM 1243-5)

Die DP-Master-Baugruppe unterstützt zusätzlich azyklische Kommunikation mithilfe von Software-Bausteinen:

- Für die Alarmbehandlung steht die Anweisung "RALRM" zur Verfügung.
- Für die Übertragung von Projektierungs- und Diagnosedaten stehen die Anweisungen "RDREC" und "WRREC" zur Verfügung.

Nicht unterstützte Funktionen des CM 1243-5:

SYNC/FREEZE  
Get\_Master\_Diag

## Weitere Kommunikationsdienste des CM 1243-5

Die DP-Master-Baugruppe CM 1243-5 unterstützt folgende weitere Kommunikationsdienste:

- S7-Kommunikation

- PUT-/GET-Dienste

Der DP-Master fungiert als Client und Server für Anfragen anderer S7-Steuerungen oder PCs über PROFIBUS.

- PG/OP-Kommunikation

Die PG-Funktionen ermöglichen das Laden von Projektierungsdaten und Anwenderprogrammen aus einem PG und die Übertragung von Diagnosedaten an ein PG.

Mögliche Kommunikationspartner für die OP-Kommunikation sind HMI-Panels, SIMATIC Panel-PCs mit WinCC flexible oder SCADA-Systeme, welche S7-Kommunikation unterstützen.

### 10.3.1.3 Weitere Eigenschaften der PROFIBUS-CMs

#### Projektierung und Baugruppentausch

Die Projektierung der Baugruppen, Netze und Verbindungen führen Sie in STEP 7 ab Version V11.0 durch.

Für die Projektierung in Fremdsystemen steht für das CM 1242-5 (DP-Slave) eine GSD-Datei auf der CD, welche zusammen mit der Baugruppe geliefert wird, und auf den Siemens Automation Customer Support-Seiten im Internet zur Verfügung.

Die Projektierungsdaten der PROFIBUS-CMs werden auf der jeweils lokalen CPU gespeichert. Dies ermöglicht im Ersatzteillfall einen einfachen Austausch dieser Kommunikationsbaugruppen.

Sie können maximal drei PROFIBUS-CMs pro Station projektieren.

### Elektrische Anschlüsse

- Spannungsversorgung
  - Das CM 1242-5 wird über den Rückwandbus der SIMATIC-Station gespeist.
  - Das CM 1243-5 besitzt einen eigenen Anschluss für die Spannungsversorgung DC 24 V.

- PROFIBUS

Die RS485-Schnittstelle des PROFIBUS-Anschlusses ist eine 9-polige Sub-D-Buchse.

Optische PROFIBUS-Netze können Sie optional über ein Optical Bus Terminal OBT oder ein Optical Link Module OLM anschließen.

### Weitergehende Informationen

Detaillierte Informationen zu den PROFIBUS-CMs enthalten die Handbücher der Geräte. Diese finden Sie im Internet auf den Seiten des Siemens Industrial Automation Customer Support unter den folgenden Beitrags-IDs.

- CM 1242-5:  
49852105 (<http://support.automation.siemens.com/WW/view/de/49852105>)
- CM 1243-5:  
49851842 (<http://support.automation.siemens.com/WW/view/de/49851842>)



### 10.3.1.4 Konfigurationsbeispiele für PROFIBUS

Im Folgenden finden Sie Beispiele für Konfigurationen, in denen das CM 1242-5 als PROFIBUS-Slave und das CM 1243-5 als PROFIBUS-Master eingesetzt wird.

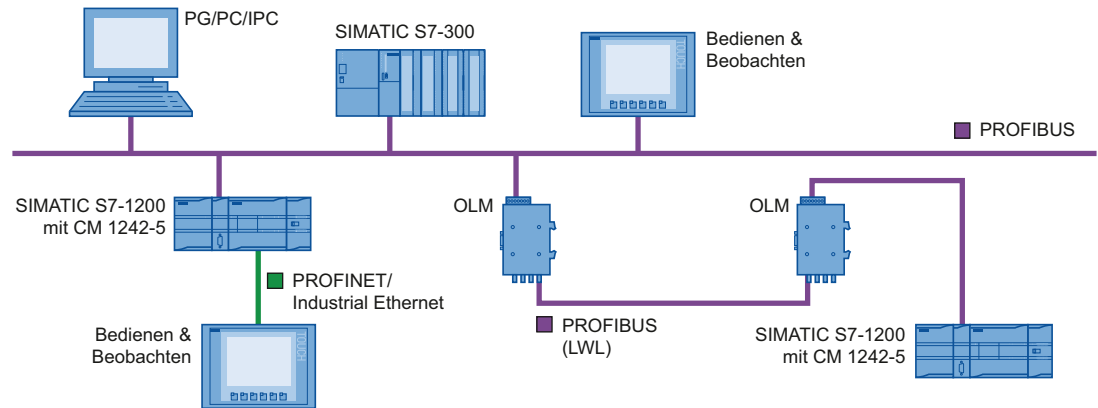


Bild 10-1 Konfigurationsbeispiel mit CM 1242-5 als PROFIBUS-Slave

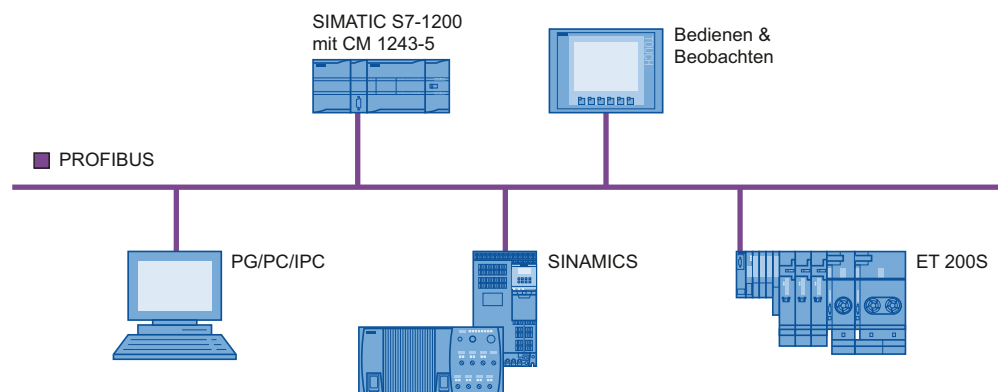


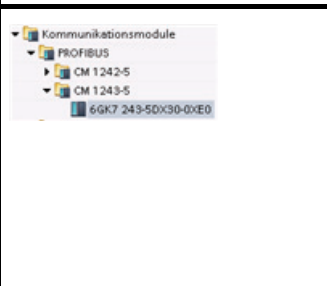
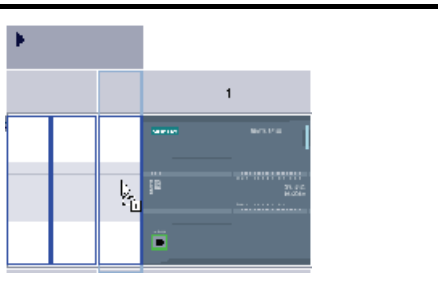
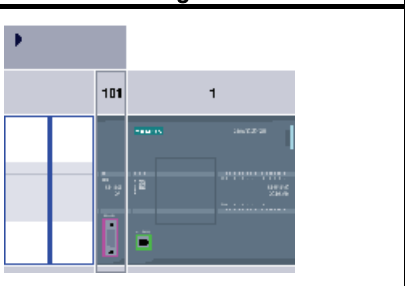
Bild 10-2 Konfigurationsbeispiel mit CM 1243-5 als PROFIBUS-Master

### 10.3.2 DP-Master und -Slave konfigurieren

#### 10.3.2.1 CM 1243-5 (DP-Master) und DP-Slave hinzufügen

Verwenden Sie im Portal "Geräte & Netze" den Hardwarekatalog, um PROFIBUS-Module zur CPU hinzuzufügen. Diese Module werden an der linken Seite der CPU angeschlossen. Um ein Modul in die Hardwarekonfiguration einzufügen, selektieren Sie das Modul im Hardwarekatalog und doppelklicken, oder Sie ziehen es in den markierten Steckplatz.

Tabelle 10- 34 PROFIBUS CM 1243-5 (DP-Master) zur Gerätekonfiguration hinzufügen

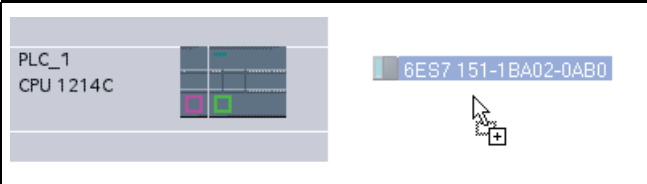

Modul	Modul auswählen	Modul einsetzen	Ergebnis
CM 1243-5 (DP-Master)			

Im Hardwarekatalog können Sie auch die DP-Slaves hinzufügen. Um beispielsweise einen ET200 S DP-Slave hinzuzufügen, erweitern Sie im Hardwarekatalog die folgenden Behälter:

- Dezentrale E/A
- ET200 S
- Schnittstellenmodule
- PROFIBUS

Wählen Sie dann "6ES7 151-1BA02-0AB0" (IM151-1 HF) in der Liste der Bestellnummern aus und fügen Sie den ET200 S DP-Slave wie in der folgenden Abbildung gezeigt ein.

Tabelle 10- 35 ET200 S DP-Slave zur Gerätekonfiguration hinzufügen

DP-Slave einfügen	Ergebnis
	

#### 10.3.2.2 Konfigurieren der logischen Netzwerkverbindungen zwischen zwei PROFIBUS-Geräten

Nachdem Sie das Modul CM 1243-5 (DP-Master) konfiguriert haben, können Sie nun Ihre Netzwerkverbindungen konfigurieren.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Um eine PROFIBUS-Verbindung anzulegen, aktivieren Sie das violettfarbene Feld (PROFIBUS) am ersten Gerät. Ziehen Sie mit der Maus eine Linie zum PROFIBUS-Feld am zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFIBUS-Verbindung hergestellt.



Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 137).

### 10.3.2.3 PROFIBUS-Adressen zum CM 1243-5 und DP-Slave zuweisen

#### Konfigurieren der PROFIBUS-Schnittstelle

Nachdem Sie die logischen Netzwerkverbindungen zwischen zwei PROFIBUS-Geräten konfiguriert haben, können Sie die Parameter für die PROFIBUS-Schnittstellen konfigurieren. Hierfür klicken Sie auf das violettfarbene PROFIBUS-Feld am Modul CM 1243-5. Daraufhin zeigt das Register "Eigenschaften" im Inspektorfenster die PROFIBUS-Schnittstelle an. Die PROFIBUS-Schnittstelle des DP-Slaves wird auf dieselbe Weise konfiguriert.

Tabelle 10- 36 PROFIBUS-Schnittstellen von CM 1243-5 (DP-Master) und ET200 S DP-Slave konfigurieren

CM 1243-5 (DP-Master)	ET200 S DP-Slave
	

① PROFIBUS-Port

#### PROFIBUS-Adresse zuweisen

In einem PROFIBUS-Netzwerk wird jedem Gerät eine PROFIBUS-Adresse zugewiesen. Diese Adresse kann im Bereich von 0 bis 127 liegen, mit folgenden Ausnahmen:

- Adresse 0: Reserviert für die Netzwerkkonfiguration und/oder an den Bus angeschlossene Programmierwerkzeuge
- Adresse 1: Reserviert von Siemens für den ersten Master
- Adresse 126: Reserviert für Geräte im Werk, die keine Schaltereinstellung haben und über das Netzwerk neu adressiert werden müssen
- Adresse 127: Reserviert für Broadcast-Meldungen an alle Geräte im Netzwerk; darf keinen betriebsfähigen Geräten zugewiesen werden

Deshalb liegen die Adressen, die für betriebsfähige PROFIBUS-Geräte verwendet werden können, im Bereich von 2 bis 125.

Wählen Sie im Eigenschaftsfenster den Eintrag "PROFIBUS-Adresse". STEP 7 zeigt den Konfigurationsdialog für die PROFIBUS-Adresse an, in dem Sie die PROFIBUS-Adresse des Geräts zuweisen.

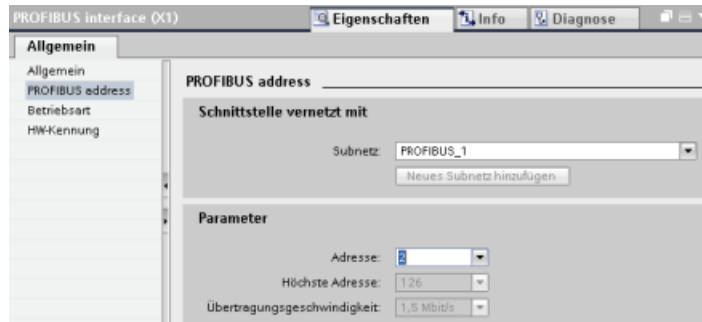


Tabelle 10- 37 Parameter für die PROFIBUS-Adresse

Parameter		Beschreibung
Subnetz	Name des Subnetzes, mit dem das Gerät verbunden ist. Um ein neues Subnetz anzulegen, klicken Sie auf Schaltfläche "Neues Subnetz hinzufügen". Die Voreinstellung ist "nicht verbunden". Zwei Arten von Verbindungen sind möglich: <ul style="list-style-type: none"> <li>Die Voreinstellung "nicht verbunden" stellt eine lokale Verbindung her.</li> <li>Wenn Ihr Netzwerk über zwei oder mehr Geräte verfügt, ist ein Subnetz erforderlich.</li> </ul>	
Parameter	Adresse	Dem Gerät zugewiesene PROFIBUS-Adresse
	Höchste Adresse	Die höchste PROFIBUS-Adresse basiert auf den aktiven Stationen auf dem PROFIBUS (z. B. DP-Master). Passive DP-Slaves haben unabhängige PROFIBUS-Adressen von 1 bis 125, auch wenn beispielsweise für die höchste PROFIBUS-Adresse 15 eingestellt ist. Die höchste PROFIBUS-Adresse ist für die Token-Weiterleitung (Weiterleitung der Senderechte) von Bedeutung, der Token wird nur an aktive Stationen weitergeleitet. Durch die Angabe der höchsten PROFIBUS-Adresse wird der Bus optimiert.
	Übertragungsgeschwindigkeit	Übertragungsgeschwindigkeit im konfigurierten PROFIBUS-Netzwerk: Die PROFIBUS-Übertragungsgeschwindigkeiten liegen zwischen 9,6 kBit/s und 12 MBit/s. Die Übertragungsgeschwindigkeit ist von den Eigenschaften der verwendeten PROFIBUS-Teilnehmer abhängig. Die Übertragungsgeschwindigkeit darf nicht größer sein als die vom langsamsten Teilnehmer unterstützte Geschwindigkeit.  Die Übertragungsgeschwindigkeit wird normalerweise für den Master im PROFIBUS-Netzwerk festgelegt, wobei alle DP-Slaves automatisch die gleiche Übertragungsgeschwindigkeit nutzen (Auto-Baud).

### 10.3.3 Anweisungen für die dezentrale E/A

Weitere Informationen zur Verwendung der Anweisungen für die dezentrale Peripherie mit diesen Kommunikationsnetzwerken finden Sie unter "Dezentrale Peripherie (PROFINET, PROFIBUS oder AS-i)" (Seite 289).

### 10.3.4 Diagnoseanweisungen

Weitere Informationen zur Verwendung dieser Anweisungen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseanweisungen" (Seite 315).

### 10.3.5 Diagnoseereignisse für die dezentrale Peripherie

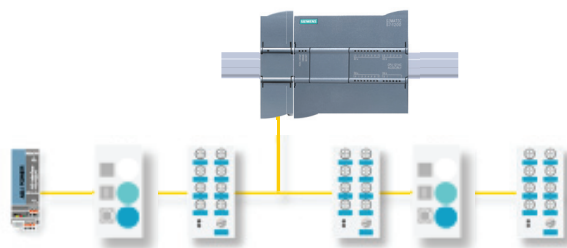
Weitere Informationen zur Verwendung dieser Diagnoseinformationen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseereignisse für die dezentrale Peripherie" (Seite 315).

## 10.4 ASi

Das S7-1200 AS-i Master CM 1243-2 ermöglicht die Anbindung eines AS-i-Netzwerks an eine S7-1200 CPU.

Die Aktor-/Sensorschnittstelle bzw. AS-i ist ein Netzwerkverbindingssystem für einen Master auf der niedrigsten Stufe in einem Automatisierungssystem. Das CM 1243-2 dient als AS-i-Master im Netzwerk. Mithilfe eines einzigen AS-i-Kabels können Sensoren und Aktoren (AS-i-Slavegeräte) über das CM 1243-2 an die CPU angeschlossen werden. Das CM 1243-2 wickelt die gesamte AS-i-Netzwerkkoordination ab und überträgt mit Hilfe der dem CM 1243-2 zugewiesenen E/A-Adressen Daten und Statusinformationen von den Aktoren und Sensoren an die CPU. Je nach Slavetyp können Sie auf binäre oder analoge Werte zugreifen. Die AS-i-Slaves sind die Eingangs- und Ausgangskanäle des AS-i-Systems und sie sind nur aktiv, wenn sie vom CM 1243-2 aufgerufen werden.

In der folgenden Abbildung ist die S7-1200 ein AS-i-Master, der das AS-i-Bedienpanel und digitale bzw. analoge Slavegeräte von E/A-Modulen steuert.



### 10.4.1 AS-i-Master und -Slavegeräte konfigurieren

Der AS-i-Master CM 1243-2 ist als Kommunikationsmodul in das S7-1200 Automatisierungssystem integriert.

Ausführliche Informationen zum AS-i-Master CM 1243-2 finden Sie im Handbuch "AS-i Master CM 1243-2 und AS-i-Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200" (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

#### 10.4.1.1 AS-i-Mastermodul CM 1243-2 und AS-i-Slave hinzufügen

Im Hardwarekatalog können Sie AS-i-Mastermodule CM1243-2 zur CPU hinzufügen. Diese Module werden an der linken Seite der CPU angeschlossen. Es können maximal drei AS-i-Mastermodule CM1243-2 verwendet werden. Um ein Modul in die Hardwarekonfiguration einzufügen, selektieren Sie das Modul im Hardwarekatalog und doppelklicken, oder Sie ziehen es in den markierten Steckplatz.

Tabelle 10- 38 AS-i-Mastermodul CM1243-2 zur Gerätekonfiguration hinzufügen

Modul	Modul auswählen	Modul einsetzen	Ergebnis
CM 1243-2 AS-i-Master			

Im Hardwarekatalog können Sie auch die AS-i-Slaves hinzufügen. Um beispielsweise einen Slave mit den Merkmalen "E/A-Modul, kompakt, digital, Eingang" hinzuzufügen, erweitern Sie im Hardwarekatalog die folgenden Behälter:

- Feldgeräte
- AS-Schnittstellen-Slaves

Wählen Sie anschließend in der Liste der Bestellnummern "3RG9 001-0AA00" (AS-i SM-U, 4DI) aus und fügen Sie den Slave mit den Merkmalen "E/A-Modul, kompakt, digital, Eingang" wie in der folgenden Abbildung gezeigt ein.

Tabelle 10- 39 AS-i-Slave zur Gerätekonfiguration hinzufügen

AS-i-Slave stecken	Ergebnis

### 10.4.1.2 Logische Netzwerkverbindungen zwischen zwei AS-i-Geräten konfigurieren

Nachdem Sie den AS-i-Master CM1243-2 konfiguriert haben, können Sie nun Ihre Netzwerkverbindungen konfigurieren.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Um eine AS-i-Verbindung anzulegen, aktivieren Sie das gelbe Feld (AS-i) am ersten Gerät. Ziehen Sie mit der Maus eine Linie zum AS-i-Feld am zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre AS-i-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 137).

### 10.4.1.3 Eigenschaften des AS-i-Masters CM1243-2 konfigurieren

Um Parameter für die AS-i-Schnittstelle zu konfigurieren, klicken Sie auf das gelbe AS-i-Feld am AS-i-Mastermodul CM1243-2. Daraufhin zeigt das Register "Eigenschaften" im Inspektorfenster die AS-i-Schnittstelle an.

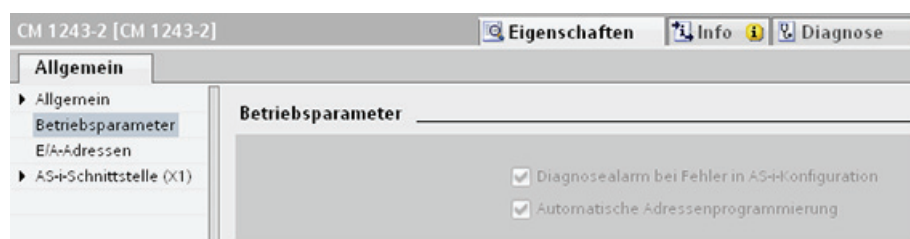
Im STEP 7-Inspektorfenster können Sie allgemeine Informationen, Adressen und Betriebsparameter anzeigen, konfigurieren und ändern:

Tabelle 10- 40 Eigenschaften des AS-i-Mastermoduls CM1243-2

Eigenschaft	Beschreibung
Allgemein	Name des AS-i-Masters CM 1243-2
Betriebsparameter	Parameter für die Antwort des AS-i-Masters
E/A-Adressen	Adressbereich für die Slave-E/A-Adressen
AS-i-Schnittstelle (X1)	Zugewiesenes AS-i-Netzwerk

#### Hinweis

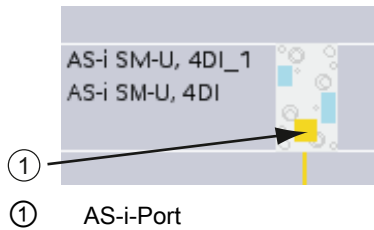
"Diagnosealarm für Fehler in der AS-i-Konfiguration" und "Automatische Adressenprogrammierung" sind stets aktiv und deshalb grau dargestellt.



### 10.4.1.4 Einem AS-i-Slave eine AS-i-Adresse zuweisen

#### AS-i-Slaveschnittstelle konfigurieren

Um Parameter für die AS-i-Schnittstelle zu konfigurieren, klicken Sie auf das gelbe AS-i-Feld am AS-i-Slave. Daraufhin zeigt das Register "Eigenschaften" im Inspektorfenster die AS-i-Schnittstelle an.



#### AS-i-Slaveadresse zuweisen

In einem AS-i-Netzwerk wird jedem Gerät eine AS-i-Slaveadresse zugewiesen. Diese Adresse kann im Bereich von 0 bis 31 liegen. Die Adresse 0 ist jedoch ausschließlich für neue Slavegeräte reserviert.

Die Slaveadressen sind 1(A oder B) bis 31(A oder B) für insgesamt maximal 62 Slavegeräte. Jede Adresse im Bereich von 1 bis 31 kann einem AS-i-Slavegerät zugewiesen werden. D. h. mit anderen Worten: Es ist nicht wichtig, ob die Slaves mit Adresse 21 beginnen oder ob der erste Slave tatsächlich die Adresse 1 erhält.

Geben Sie hier die AS-i-Slaveadresse ein.

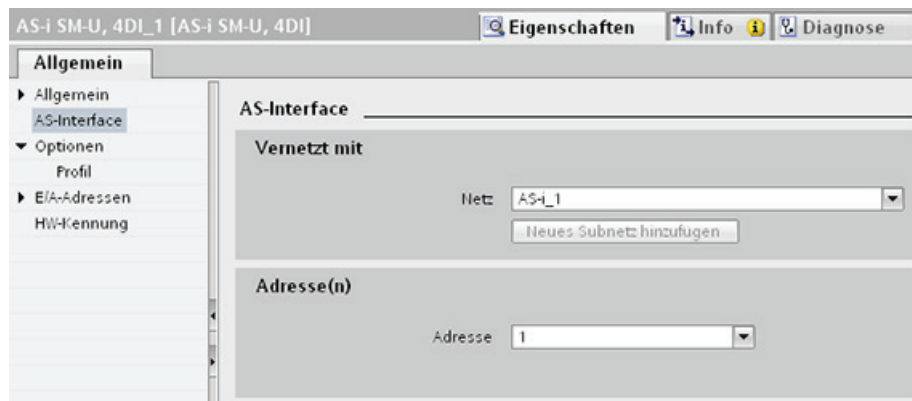


Tabelle 10- 41 Parameter für die AS-i-Schnittstelle

Parameter	Beschreibung
Netzwerk	Name des Netzwerks, mit dem das Gerät verbunden ist.
Adresse(n)	Zugewiesene AS-i-Adresse für das Slavegerät im Bereich von 1(A oder B) bis 31(A oder B) für insgesamt bis zu 62 Slavegeräte

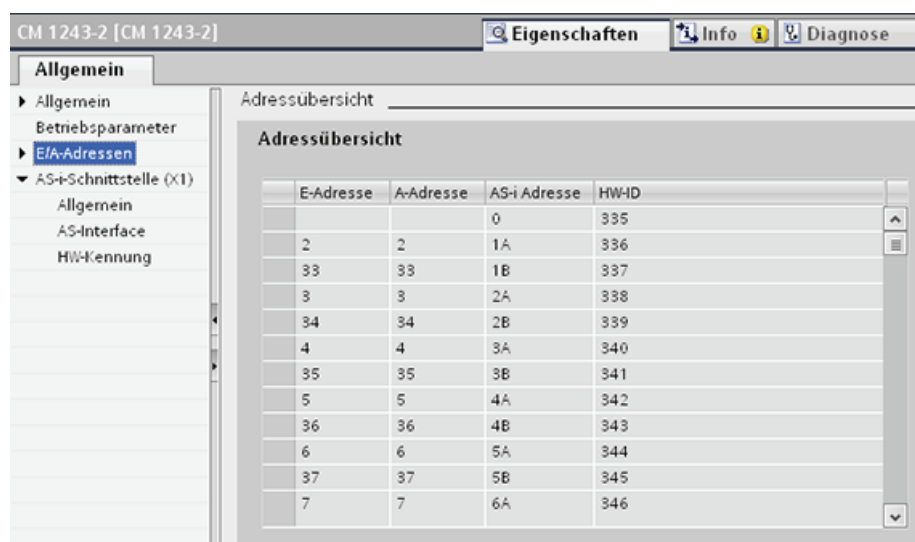


## 10.4.2 Datenaustausch zwischen dem Anwenderprogramm und AS-i-Slaves

### 10.4.2.1 STEP 7 Basic konfigurieren

Der AS-i-Master reserviert einen 62 Byte großen Datenbereich im E/A-Bereich der CPU. Der Zugriff auf die Digitaldaten erfolgt hier in Byte. Für jeden Slave gibt es ein Byte Eingangs- und ein Byte Ausgangsdaten.

Die Zuweisung der AS-i-Verbindungen der digitalen AS-i-Slaves zu den Datenbits des zugewiesenen Bytes wird im Inspektorfenster des AS-i-Mastermoduls CM 1243-2 angezeigt.



E-Adresse	A-Adresse	AS-i Adresse	HW-ID
		0	335
2	2	1A	336
33	33	1B	337
3	3	2A	338
34	34	2B	339
4	4	3A	340
35	35	3B	341
5	5	4A	342
36	36	4B	343
6	6	5A	344
37	37	5B	345
7	7	6A	346

Sie können auf die Daten der AS-i-Slaves im Anwenderprogramm zugreifen, indem Sie die angezeigten E/A-Adressen mit den entsprechenden Bitverknüpfungsoperationen (z. B. "UND") oder Bitzuweisungen verwenden.

#### Hinweis

Wenn Sie die AS-i-Slaves nicht mit STEP 7 konfigurieren, wird die "Systemzuweisung" automatisch aktiviert.

Wenn Sie keine Slaves konfigurieren, müssen Sie das AS-i-Master CM1243-2 über die tatsächliche Buskonfiguration informieren. Hierfür verwenden Sie die Online-Funktion "IST > SOLL".

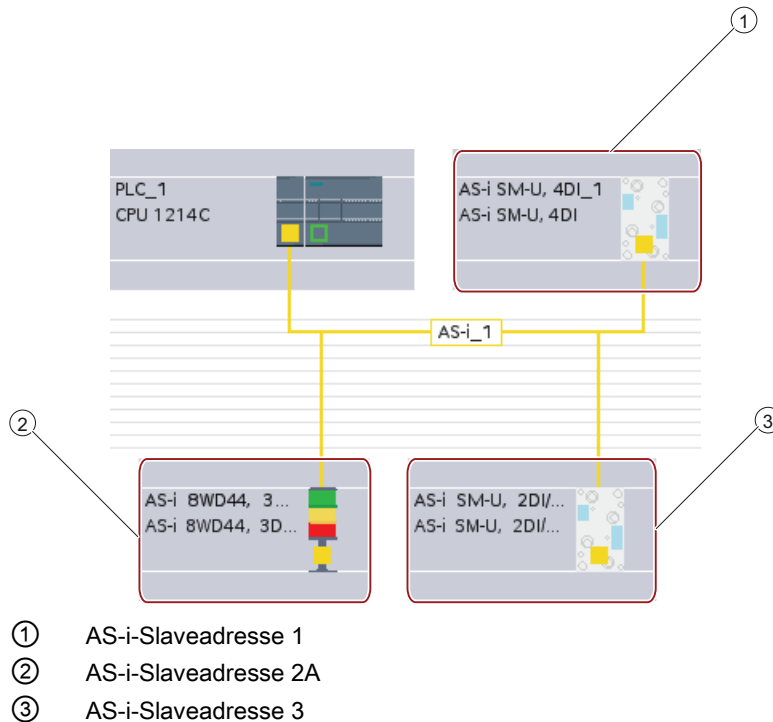
#### Weitere Informationen

Ausführliche Informationen zum AS-i-Master CM 1243-2 finden Sie im Handbuch "AS-i Master CM 1243-2 und AS-i-Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200" (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

### 10.4.2.2 Slaves mit STEP 7 konfigurieren

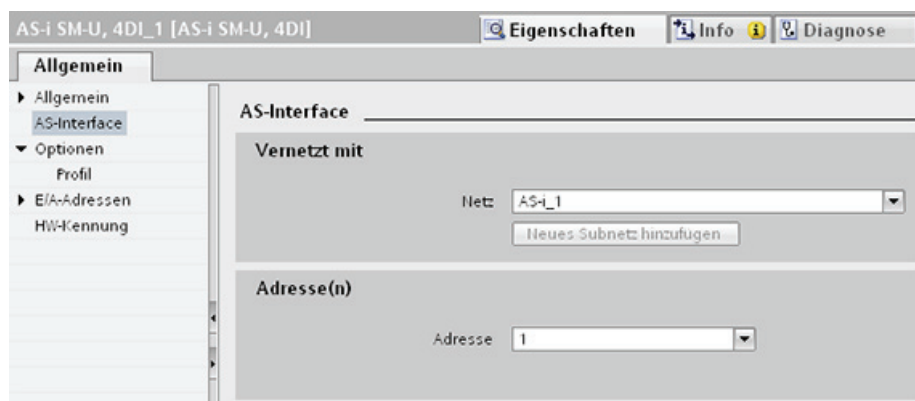
#### Digitale AS-i-Werte übertragen

Die CPU greift im zyklischen Betrieb über den AS-i-Master CM1243-2 auf die digitalen Eingänge und Ausgänge der AS-i-Slaves zu. Der Zugriff auf die Daten erfolgt über E/A-Adressen oder mittels Datensatzübertragung.

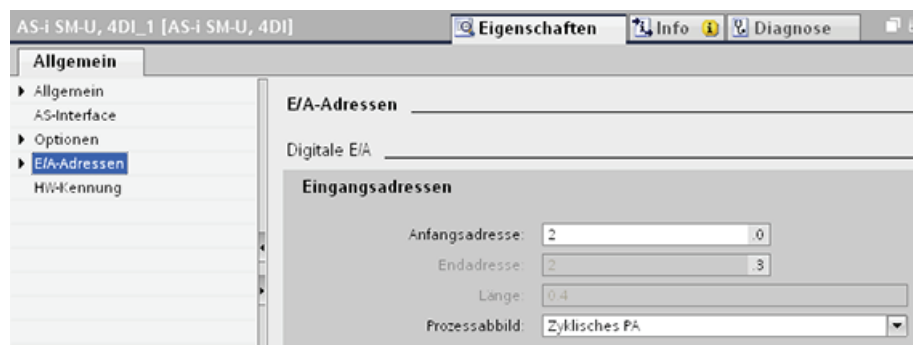


Der Zugriff auf die Digitaldaten erfolgt hier in Bytes (mit anderen Worten: jedem digitalen AS-i-Slave ist ein Byte zugeordnet). Wenn Sie die AS-i-Slaves in STEP 7 konfigurieren, wird für den jeweiligen AS-i-Slave die E/A-Adresse für den Zugriff auf die Daten aus dem Anwenderprogramm im Inspektorfenster angezeigt.

Dem Digitaleingabemodul (AS-i SM-U, 4DI) im AS-i-Netzwerk oben wurde die Slaveadresse 1 zugewiesen. Wenn Sie auf das Digitaleingabemodul klicken, wird in den "Eigenschaften" des Geräts im Register "AS-Schnittstelle" die Slaveadresse wie im Folgenden angezeigt:

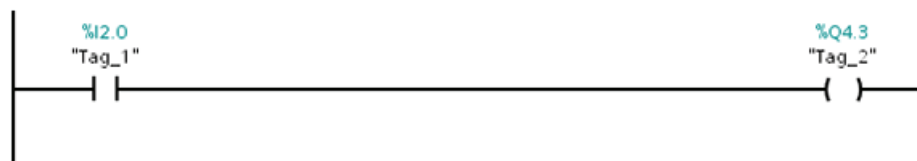


Dem Digitaleingabemodul (AS-i SM-U, 4DI) im AS-i-Netzwerk oben wurde die E/A-Adresse 2 zugewiesen. Wenn Sie auf das Digitaleingabemodul klicken, wird in den "Eigenschaften" des Geräts im Register "E/A-Adressen" die E/A-Adresse wie im Folgenden angezeigt:



Sie können auf die Daten der AS-i-Slaves im Anwenderprogramm zugreifen, indem Sie die deren E/A-Adressen mit den entsprechenden Bitverknüpfungsoperationen (z. B. "UND") oder Bitzuweisungen verwenden. Das folgende einfache Programm zeigt die Funktionsweise der Zuweisung:

In diesem Programm wird Eingang 2.0 abgefragt. Im AS-i-System gehört dieser Eingang zu Slave 1 (Eingangsbyte 2, Bit 0). Ausgang 4.3, der dann gesetzt wird, entspricht dem AS-i-Slave 3 (Ausgangsbyte 4, Bit 3).



## Analoge AS-i-Werte übertragen

Wenn Sie einen AS-i-Slave in STEP 7 als analogen Slave konfiguriert haben, können Sie über das Prozessabbild der CPU auf die analogen Daten dieses AS-i-Slaves zugreifen.

Wenn Sie den analogen Slave nicht in STEP 7 konfiguriert haben, können Sie nur über die azyklischen Funktionen (Datensatzschnittstelle) auf die Daten des AS-i-Slaves zugreifen. Im Anwenderprogramm der CPU werden AS-i-Aufrufe über die Anweisungen für die dezentrale Peripherie, nämlich RDREC (Datensatz lesen) und WRREC (Datensatz schreiben), gelesen und geschrieben.

### Hinweis

Eine über STEP 7 angegebene und in die S7-Station geladene Konfiguration der AS-i-Slaves wird beim Anlauf der S7-Station von der CPU in den AS-i-Master CM1243-2 geladen. Eine ggf. bereits vorhandene Konfiguration, die über die Online-Funktion der "Systemzuweisung" (Seite 513) ("IST -> SOLL") bestimmt wurde, wird überschrieben.

**Weitere Informationen**

Ausführliche Informationen zum AS-i-Master CM 1243-2 finden Sie im Handbuch "AS-I Master CM 1243-2 und AS-i-Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200" (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

**10.4.3 Anweisungen für die dezentrale E/A**

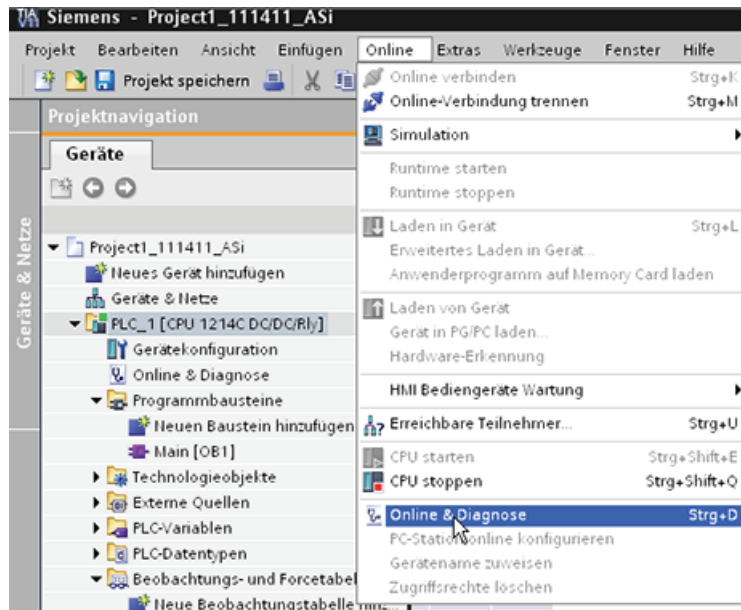
Weitere Informationen zur Verwendung der Anweisungen für die dezentrale Peripherie mit diesen Kommunikationsnetzwerken finden Sie unter "Dezentrale Peripherie (PROFINET, PROFIBUS oder AS-i)" (Seite 289).

**10.4.4 Mit AS-i-Online-Werkzeugen arbeiten**

**Betriebsarten der AS-i online ändern**

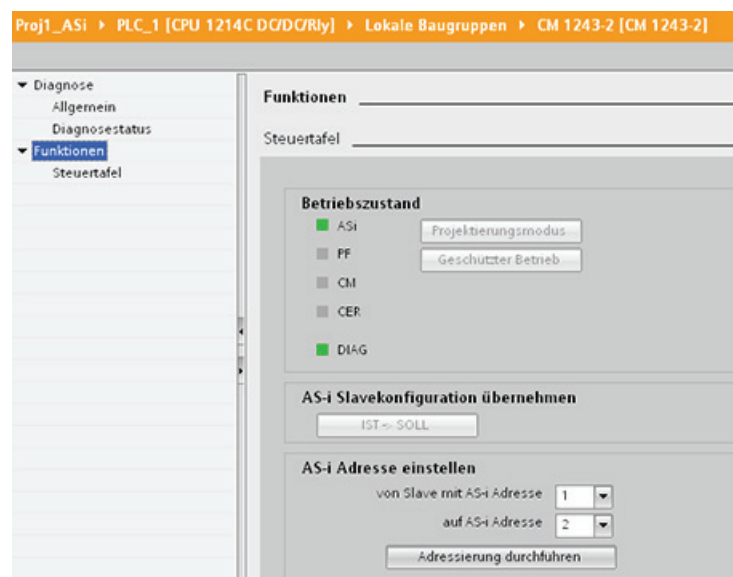
Sie müssen online gehen, um die Betriebsarten der AS-i anzuzeigen und zu ändern.

Um online zu gehen, müssen Sie zunächst das AS-i-Mastermodul CM1243-2 auswählen und die "Gerätekonfiguration" aufrufen. Klicken Sie dann in der Symbolleiste auf die Schaltfläche "Online gehen". Wählen Sie dann im Menü "Online" den Befehl "Online und Diagnose".



Es gibt zwei AS-i-Betriebsarten:

- Schutzmodus:
  - Das AS-i-Slavegerät und die E/A-Adressen der CPU können nicht geändert werden.
  - Die grüne LED "CM" ist ausgeschaltet.
- Konfigurationsmodus:
  - Sie können in Ihrem AS-i-Slavegerät und an den E/A-Adressen der CPU erforderliche Änderungen vornehmen.
  - Die grüne LED "CM" ist eingeschaltet.



Die AS-i-Slaveadresse können Sie im Feld "AS-i-Adresse festlegen" ändern. Ein neuer Slave, dem noch keine Adresse zugewiesen wurde, hat immer die Adresse 0. Er wird vom Master als neuer Slave ohne Adresszuweisung erkannt und wird erst nach Zuweisung einer Adresse in die normale Kommunikation einbezogen.

## Konfigurationsfehler

Wenn die gelbe LED "CER" leuchtet, liegt ein Fehler in der Gerätekonfiguration des AS-i-Slaves vor. Wählen Sie die Schaltfläche "IST > SOLL", um die Slave-Gerätekonfiguration des AS-i-Mastermoduls CM1243-2 durch die Slave-Gerätekonfiguration des AS-i-Feldnetzwerks zu überschreiben.

## 10.5 S7-Kommunikation

### 10.5.1 Anweisungen GET und PUT

Mit den Anweisungen GET und PUT können Sie mit S7-CPU's über PROFINET- und PROFIBUS-Verbindungen kommunizieren:

- Zugriff auf Daten in einer dezentralen CPU: Eine S7-1200 CPU kann im Eingabefeld ADDR\_x nur absolute Adressen verwenden, um Variablen dezentraler CPU's (S7-200/300/400/1200) anzusprechen.
- Zugriff auf Daten in einem Standard-DB: Eine S7-1200 CPU kann im Eingabefeld ADDR\_x nur absolute Adressen verwenden, um DB-Variablen in einem Standard-DB einer dezentralen S7-CPU anzusprechen.
- Zugriff auf Daten in einem optimierten DB: Eine S7-1200 CPU kann nicht auf DB-Variablen in einem optimierten DB einer dezentralen S7-1200 CPU zugreifen.
- Zugriff auf Daten in einer lokalen CPU: Eine S7-1200 CPU kann entweder absolute Adressen oder symbolische Adressen als Eingabe im Eingabefeld RD\_x oder SD\_x der Anweisung GET oder PUT verwenden.

Tabelle 10- 42 Anweisungen GET und PUT

KOP/FUP	SCL	Beschreibung
	<pre>"GET_DB" (     req:= _bool_in_,     ID:= _word_in_,     ndr=&gt; _bool_out_,     error=&gt; _bool_out_,     status=&gt; _word_out_,     addr_1:= _remote_inout_,     [...addr_4:= _remote_inout_,]     rd_1:= _variant_inout_     [...rd_4:= _variant_inout_]);</pre>	<p>Mit der Anweisung GET lesen Sie Daten aus einer entfernten S7-CPU aus. Die entfernte CPU kann sich dabei im Betriebszustand RUN oder STOP befinden.</p> <p>STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.</p>
	<pre>"PUT_DB" (     req:= _bool_in_,     ID:= _word_in_,     done=&gt; _bool_out_,     error=&gt; _bool_out_,     status=&gt; _word_out_,     addr_1:= _remote_inout_,     [...addr_4:= _remote_inout_,]     sd_1:= _variant_inout_,     [...sd_4:= _variant_inout_]);</pre>	<p>Mit der Anweisung PUT schreiben Sie Daten in eine entfernte S7-CPU. Die entfernte CPU kann sich dabei im Betriebszustand RUN oder STOP befinden.</p> <p>STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.</p>

Tabelle 10- 43 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	Input	Bool Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet.
ID	Input	CONN_PRG (Word) S7 Verbindungs-ID (Hex)
NDR (GET)	Output	Bool Neue Daten bereit: <ul style="list-style-type: none"> <li>0: Die Anforderung wurde noch nicht gestartet oder läuft noch.</li> <li>1: Der Auftrag wurde erfolgreich beendet.</li> </ul>
DONE (PUT)	Output	Bool DONE: <ul style="list-style-type: none"> <li>0: Die Anforderung wurde noch nicht gestartet oder läuft noch.</li> <li>1: Der Auftrag wurde erfolgreich beendet.</li> </ul>

Parameter und Datentyp		Datentyp	Beschreibung
ERROR STATUS	Output Output	Bool Word	<ul style="list-style-type: none"> <li>ERROR=0 STATUS-Wert:                             <ul style="list-style-type: none"> <li>0000H: Weder Warnung noch Fehler.</li> <li>&lt;&gt; 0000H: Warnung, STATUS liefert ausführliche Informationen.</li> </ul> </li> <li>ERROR=1 Fehler. STATUS liefert ausführliche Informationen über die Natur des Fehlers.</li> </ul>
ADDR_1	InOut	Entfernt	Pointer auf die Speicherbereiche in der entfernten CPU, die die zu lesenden (GET) oder zu sendenden (PUT) Daten speichert.
ADDR_2	InOut	Entfernt	
ADDR_3	InOut	Entfernt	
ADDR_4	InOut	Entfernt	
RD_1 (GET) SD_1 (PUT)	InOut	Variante	Pointer auf die Speicherbereiche in der lokalen CPU, die die zu lesenden (GET) oder zu sendenden (PUT) Daten speichert.
RD_2 (GET) SD_2 (PUT)	InOut	Variante	Zulässige Datentypen: Bool (nur ein Bit zulässig), Byte, Char, Word, Int, DWord, DInt oder Real.
RD_3 (GET) SD_3 (PUT)	InOut	Variante	Hinweis: Wenn der Pointer auf einen DB zugreift, müssen Sie die absolute Adresse angeben, z. B.:
RD_4 (GET) SD_4 (PUT)	InOut	Variante	P# DB10.DBX5.0 Byte 10 In diesem Fall stellt der Wert 10 die Anzahl von Bytes für die Anweisung GET oder PUT dar.

Sie müssen sicherstellen, dass Länge (Anzahl der Bytes) und Datentypen der Parameter ADDR\_x (entfernte CPU) und RD\_x oder SD\_x (lokale CPU) übereinstimmen. Die Zahl nach der Kennung "Byte" ist die Anzahl von Bytes, die vom Parameter ADDR\_x, RD\_x oder SD\_x angegeben wird.

**Hinweis**

Die Gesamtzahl der von einer Anweisung GET empfangenen Bytes bzw. die Gesamtzahl der von einer Anweisung PUT gesendeten Bytes ist begrenzt. Die Begrenzung richtet sich danach, wie viele der vier möglichen Adress- und Speicherbereiche Sie verwenden:

- Wenn Sie nur ADDR\_1 und RD\_1/SD\_1 verwenden, kann eine Anweisung GET 222 Bytes abrufen und eine Anweisung PUT 212 Bytes senden.
- Wenn Sie ADDR\_1, RD\_1/SD\_1, ADDR\_2 und RD\_2/SD\_2 verwenden, kann eine Anweisung GET insgesamt 218 Bytes abrufen und eine Anweisung PUT insgesamt 196 Bytes senden.
- Wenn Sie ADDR\_1, RD\_1/SD\_1, ADDR\_2, RD\_2/SD\_2, ADDR\_3 und RD\_3/SD\_3 verwenden, kann eine Anweisung GET insgesamt 214 Bytes abrufen und eine Anweisung PUT insgesamt 180 Bytes senden.
- Wenn Sie ADDR\_1, RD\_1/SD\_1, ADDR\_2, RD\_2/SD\_2, ADDR\_3, RD\_3/SD\_3, ADDR\_4, RD\_4/SD\_4 verwenden, kann eine Anweisung GET insgesamt 210 Bytes abrufen und eine Anweisung PUT insgesamt 164 Bytes senden.

Die Summe der Byteanzahlen von jeder Ihrer Adress- und Speicherbereichsparameter muss kleiner oder gleich den festgelegten Grenzwerten sein. Wenn Sie diese Grenzwerte überschreiten, gibt die Anweisung GET oder PUT einen Fehler aus.



Bei der steigenden Flanke am Parameter REQ lädt die Leseanweisung (GET) oder die Schreibanweisung (PUT) die Parameter ID, ADDR\_1 und RD\_1 (GET) oder SD\_1 (PUT).

- Bei GET: Die entfernte CPU gibt die angeforderten Daten an die Empfangsbereiche aus (RD\_x). Sie beginnt damit im nächsten Zyklus. Wenn die Leseanweisung fehlerfrei beendet wurde, wird der Parameter NDR auf 1 gesetzt. Eine neue Anweisung kann erst gestartet werden, wenn die vorherige Anweisung beendet ist.
- Bei PUT: Die lokale CPU beginnt, die Daten (SD\_x) an die Adresse (ADDR\_x) im Speicher der entfernten CPU zu senden. Wenn die Schreibanweisung fehlerfrei beendet wurde, gibt die entfernte CPU eine Ausführungsquittierung aus. Der Parameter DONE der Anweisung PUT wird dann auf 1 gesetzt. Eine neue Schreibanweisung kann erst gestartet werden, wenn die vorherige Anweisung beendet ist.

#### Hinweis

Um Datenkonsistenz sicherzustellen, prüfen Sie stets, ob die Anweisung beendet wurde (NDR = 1 bei GET bzw. DONE = 1 bei PUT), bevor Sie auf die Daten zugreifen oder eine weitere Lese- oder Schreibanweisung ausführen.

Die Parameter ERROR und STATUS liefern Informationen zum Zustand der Lese- (GET) oder Schreibanweisung (PUT).

Tabelle 10- 44 Fehlerinformation

ERROR	STATUS (dezimal)	Beschreibung
0	11	<ul style="list-style-type: none"> <li>• Der neue Auftrag kann nicht wirksam werden, weil der vorherige Auftrag noch nicht beendet ist.</li> <li>• Der Auftrag wird nun in einer Prioritätsklasse mit geringerer Priorität verarbeitet.</li> </ul>
0	25	Kommunikation gestartet. Auftrag wird bearbeitet.
1	1	Kommunikationsprobleme, zum Beispiel: <ul style="list-style-type: none"> <li>• Verbindungsbeschreibung nicht geladen (lokal oder entfernt)</li> <li>• Verbindung unterbrochen (Beispiel: Kabel, CPU ausgeschaltet oder CM/CB/CP ist in STOP)</li> <li>• Verbindung zum Partner noch nicht aufgebaut</li> </ul>
1	2	Negative Quittierung vom Partnergerät. Die Aufgabe kann nicht ausgeführt werden.
1	4	Fehler in den Sendebereichs-Pointern (RD_x bei GET, SD_x bei PUT) bezüglich Datenlänge oder Datentyp.
1	8	Zugriffsfehler auf der Partner-CPU
1	10	Zugriff auf den lokalen Anwenderspeicher nicht möglich (Beispiel: Zugriffsversuch auf gelöschten DB)
1	12	Als der SFB aufgerufen wurde: <ul style="list-style-type: none"> <li>• Wurde ein Instanz-DB angegeben, der nicht zu GET oder PUT gehört.</li> <li>• Wurde kein Instanz-DB angegeben, sondern ein gemeinsam genutzter DB</li> <li>• Wurde kein Instanz-DB gefunden (Laden eines neuen Instanz-DB).</li> </ul>

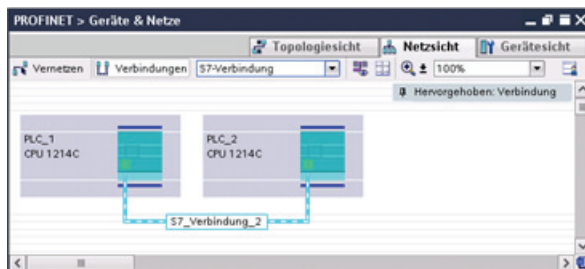
ERROR	STATUS (dezimal)	Beschreibung
1	20	<ul style="list-style-type: none"> <li>• Maximale Anzahl paralleler Aufträge/Instanzen überschritten</li> <li>• Die Instanzen wurden bei CPU-RUN überladen</li> </ul> Dieser Zustand ist während der ersten Ausführung der Anweisung GET oder PUT möglich.
1	27	Es gibt keine entsprechende Anweisung GET oder PUT in der CPU.

### 10.5.2 S7-Verbindung erstellen

Die ausgewählte Verbindungsart baut eine Kommunikationsverbindung zu einer Partnerstation auf. Die Verbindung wird eingerichtet, aufgebaut und automatisch überwacht.

Im Portal "Geräte & Netze" können Sie in der "Netzsicht" die Geräte in Ihrem Projekt vernetzen. Klicken Sie zunächst auf das Register "Verbindungen" und wählen Sie dann über die Klappliste rechts den Verbindungstyp aus (z. B. eine S7-Verbindung). Klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen Sie eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Erstellen einer Netzwerkverbindung" (Seite 137).



Klicken Sie auf die Schaltfläche "Hervorgehoben: Verbindung", um den Konfigurationsdialog "Eigenschaften" der Kommunikationsanweisung aufzurufen.

### 10.5.3 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

#### Allgemeine Parameter konfigurieren

Sie legen die Kommunikationsparameter im Dialog "Eigenschaften" der Kommunikationsanweisung fest. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung selektiert haben.

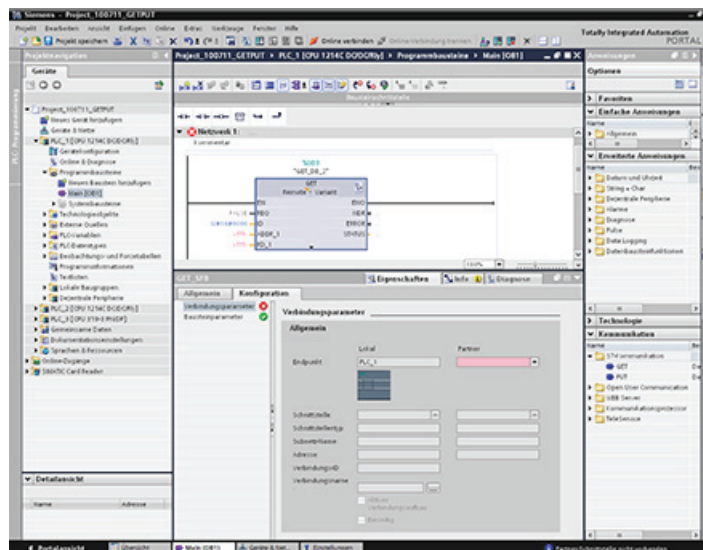
Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 138)".

Die zu verwendenden TSAPs oder Ports definieren Sie im Dialog "Verbindungsparameter" unter "Adressdetails". Der TSAP oder Port einer Verbindung in der CPU wird im Feld "Lokaler TSAP" eingegeben. Der TSAP oder Port für die Verbindung in der Partner-CPU wird im Feld "Partner-TSAP" eingegeben.

## 10.5.4 Zuweisung von Verbindungsparametern für GET/PUT

Die Zuweisung der Verbindungsparameter für die GET/PUT-Anweisungen ist eine Hilfe für den Anwender bei der Konfiguration der S7-Verbindungen für die S7-Kommunikation von CPU zu CPU.

Nach dem Einfügen eines GET- oder PUT-Bausteins wird die Zuweisung der Verbindungsparameter für die GET/PUT-Anweisungen gestartet:



Im Inspektorenfenster werden die Eigenschaften der Verbindung angezeigt, wenn Sie einen Teil der Anweisung auswählen. Sie können die Kommunikationsparameter über das Register "Konfiguration" im Dialog "Eigenschaften" der Kommunikationsanweisung festlegen.

### 10.5.4.1 Verbindungsparameter

Auf der Seite "Verbindungsparameter" können Sie die erforderliche S7-Verbindung und den Parameter "Verbindungs-ID" konfigurieren, der vom Parameter "ID" des GET/PUT-Bausteins referenziert wird. Der Inhalt der Seite umfasst Informationen zum lokalen Endpunkt und Sie können zudem die lokale Schnittstelle festlegen. Ferner können Sie den Partner-Endpunkt definieren.

Auf der Seite "Bausteinparameter" können Sie weitere Bausteinparameter konfigurieren.

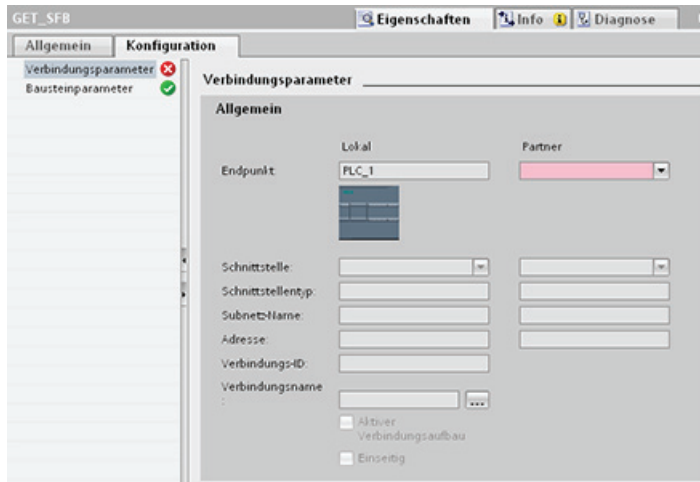


Tabelle 10- 45 Verbindungsparameter: Allgemeine Definitionen

Parameter	Definition	
Verbindungsparameter: Allgemein	Endpunkt	"Lokaler Endpunkt": Name der lokalen CPU "Partner-Endpunkt": Name der Partner-CPU (dezentral) Hinweis: In der Klappliste "Partner-Endpunkt" zeigt das System alle potenziellen S7-Verbindungspartner des aktuellen Projekts sowie die Option "Nicht spezifiziert" an. Ein nicht spezifizierter Partner ist ein Kommunikationspartner, der sich gegenwärtig nicht im STEP 7-Projekt befindet (z. B. ein Kommunikationspartner eines Fremdherstellers).
	Schnittstelle	Name der Schnittstellen Hinweis: Sie können die Verbindung ändern, indem Sie die lokalen und Partnerschnittstellen ändern.
	Schnittstellentyp	Typ der Schnittstelle
	Subnetzname	Name der Subnetze
	Adresse	Zugewiesene IP-Adressen Hinweis: Für einen "nicht spezifizierten" Kommunikationspartner können Sie die entfernte Adresse eines Fremdgeräts angeben.
	Verbindungs-ID	ID-Nummer: Wird von der Parameterzuweisung der GET/PUT-Verbindung automatisch generiert
	Verbindungsname	Datenspeicheradresse für die lokale CPU und die Partner-CPU: Wird von der Parameterzuweisung der GET/PUT-Verbindung automatisch generiert
	Aktiver Verbindungsaufbau	Kontrollkästchen zur Auswahl der lokalen CPU als aktive Verbindung
	Unidirektional	Kontrollkästchen, um eine unidirektionale oder eine bidirektionale Verbindung anzugeben; schreibgeschützt Hinweis: In einer GET/PUT-Verbindung mit PROFINET können sowohl das lokale Gerät als auch das Partnergerät als Server oder Client fungieren. Dies ermöglicht eine bidirektionale Verbindung und das Kontrollkästchen "Unidirektional" ist deaktiviert. Bei einer GET/PUT-Verbindung mit PROFIBUS kann das Partnergerät in einigen Fällen nur als Server fungieren (z. B. bei einer S7-300) und das Kontrollkästchen "Unidirektional" ist aktiviert.

## Parameter Verbindungs-ID

Es gibt drei Möglichkeiten, die systemdefinierten Verbindungs-IDs zu ändern:

1. Sie können die aktuelle ID direkt im GET/PUT-Baustein ändern. Wenn die neue ID zu einer bereits vorhandenen Verbindung gehört, wird die Verbindung geändert.
2. Sie können die aktuelle ID direkt im GET/PUT-Baustein ändern, doch die neue ID ist noch nicht vorhanden. Das System erstellt eine neue S7-Verbindung.
3. Sie können die aktuelle ID über den Dialog "Verbindungsübersicht" ändern: Ihre Eingabe wird mit dem ID-Parameter des entsprechenden GET/PUT-Bausteins synchronisiert.

---

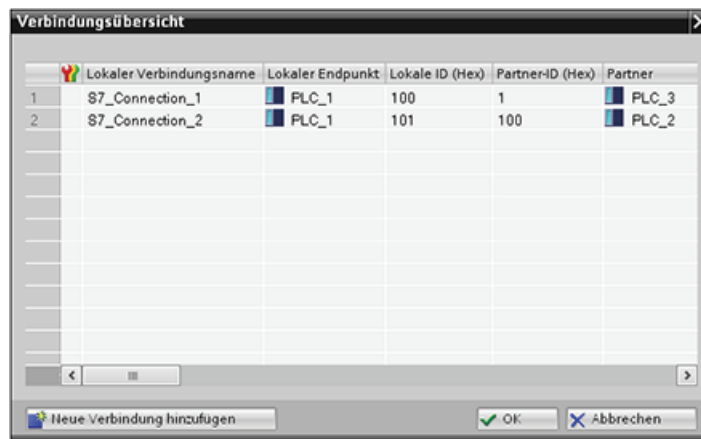
### Hinweis

Der Parameter "ID" des GET/PUT-Bausteins ist kein Verbindungsname, sondern ein numerischer Ausdruck, der wie im folgenden Beispiel geschrieben wird: W#16#1

---

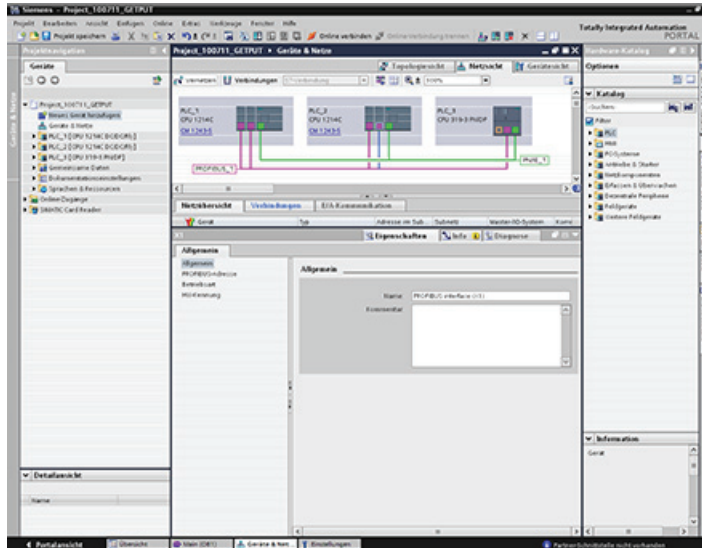
## Parameter Verbindungsname

Der Verbindungsname lässt sich über ein besonderes Bedienelement, den Dialog "Verbindungsübersicht", bearbeiten. In diesem Dialog werden alle verfügbaren S7-Verbindungen angezeigt, die als Alternative für die aktuelle GET/PUT-Kommunikation ausgewählt werden können. Sie können in dieser Tabelle eine vollständig neue Verbindung anlegen. Klicken Sie auf die Schaltfläche rechts vom Feld "Verbindungsname", um den Dialog "Verbindungsübersicht" aufzurufen.



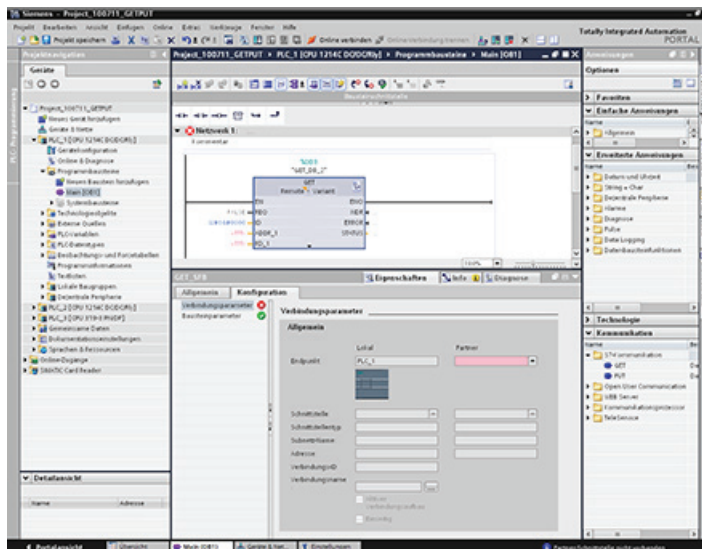
### 10.5.4.2 S7-Verbindung von CPU zu CPU konfigurieren

Bei einer Konfiguration von PLC\_1, PLC\_2 und PLC\_3, wie in der folgenden Abbildung gezeigt, fügen Sie GET- oder PUT-Bausteine für "PLC\_1" ein.



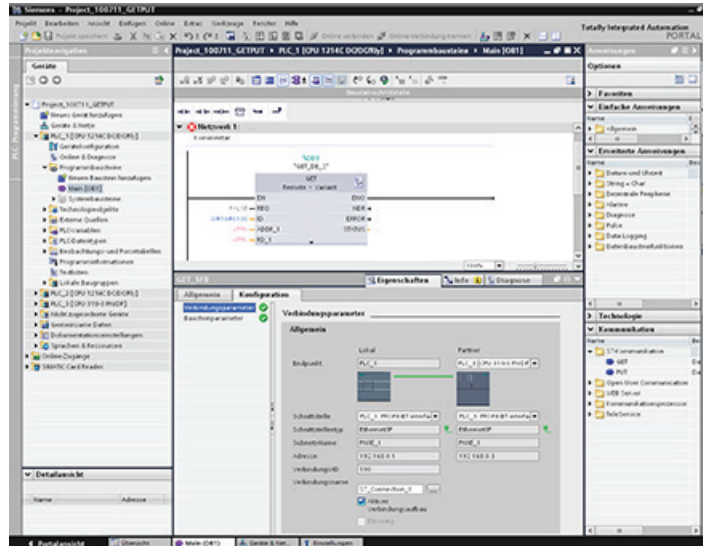
Für die Anweisung GET oder PUT wird im Inspektorfenster automatisch das Register "Eigenschaften" mit den folgenden Menüoptionen angezeigt:

- "Konfiguration"
- "Verbindungsparameter"



## PROFINET S7-Verbindung konfigurieren

Wählen Sie für den "Partner-Endpunkt" die Option "PLC\_3" aus.



Das System reagiert mit den folgenden Änderungen:

Tabelle 10- 46 Verbindungsparameter: Allgemeine Werte

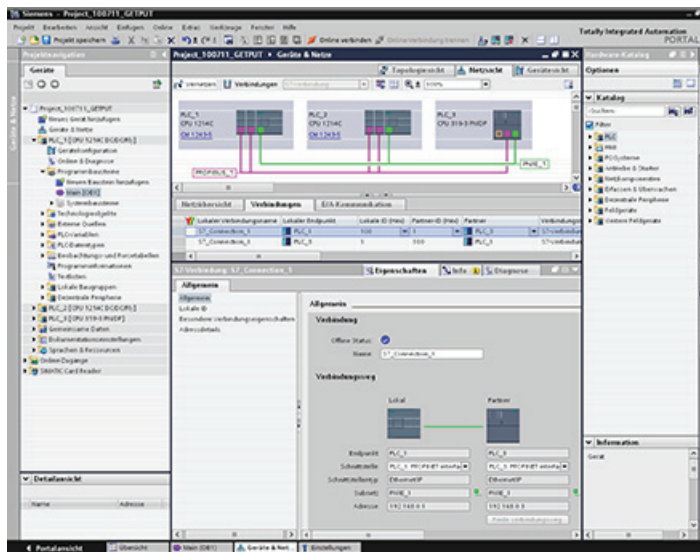
Parameter	Definition
Verbindungsparameter: Allgemein	<b>Endpunkt</b> "Lokaler Endpunkt" enthält "PLC_1" (schreibgeschützt). Das Feld "Partner-Endpunkt" enthält "PLC_3[CPU319-3PN/DP]": <ul style="list-style-type: none"> <li>Die Farbe wechselt von Rot nach Weiß.</li> <li>Das "Partner"-Gerätebild wird angezeigt.</li> <li>Eine Verbindungslinie erscheint zwischen den PLC_1- und PLC_3-Gerätebildern (grüne Ethernet-Linie).</li> </ul>
	<b>Schnittstelle</b> "Lokale Schnittstelle" enthält "CPU1214C DC/DC/DC, PROFINET-Schnittstelle (R0/S1)". "Partner-Schnittstelle" enthält: "CPU319-3PN/DP, PROFINET-Schnittstelle (R0/S2)".
	<b>Schnittstellentyp</b> "Lokaler Schnittstellentyp" enthält "Ethernet/IP"; die Bedienung ist schreibgeschützt. "Partner-Schnittstellentyp" enthält "Ethernet/IP"; die Bedienung ist schreibgeschützt. Die Schnittstellentypbilder werden rechts neben dem lokalen und dem Partner-Schnittstellentyp angezeigt (grünes Ethernet-Symbol).
	<b>Subnetzname</b> "Lokaler Subnetzname" enthält "PN/IE_1"; die Bedienung ist schreibgeschützt. "Partner-Subnetzname" enthält "PN/IE_1"; die Bedienung ist schreibgeschützt.
	<b>Adresse</b> "Lokale Adresse" enthält die lokale IP-Adresse; die Bedienung ist schreibgeschützt. "Partneradresse" enthält die Partner-IP-Adresse; die Bedienung ist schreibgeschützt.

Parameter	Definition
Verbindungs-ID	"Verbindungs-ID" enthält "100". Im Hauptprogramm [OB1] des Programmiereditors enthält die "Verbindungs-ID" des GET/PUT-Bausteins ebenfalls den Wert "100".
Verbindungsname	"Verbindungsname" enthält den Standardverbindungsname (Beispiel: "S7_Verbindung_1"); die Bedienung ist aktiviert.
Aktiver Verbindungsaufbau	Aktiviert, um die lokale CPU als aktive Verbindung auszuwählen.
Unidirektional	Schreibgeschützt und nicht aktiviert. Hinweis: "PLC_1" (eine S7-1200 CPU 1214CDC/DC/RIs) und "PLC_3" (eine S7-300 CPU 319-3PN/DP) können in einer PROFINET GET/PUT-Verbindung beide als Server und als Client fungieren, so dass eine bidirektionale Verbindung möglich ist.

Das GET/PUT-Symbol in der Eigenschaftsansicht wechselt von Rot nach Grün.

### Vollständige PROFINET S7-Verbindung

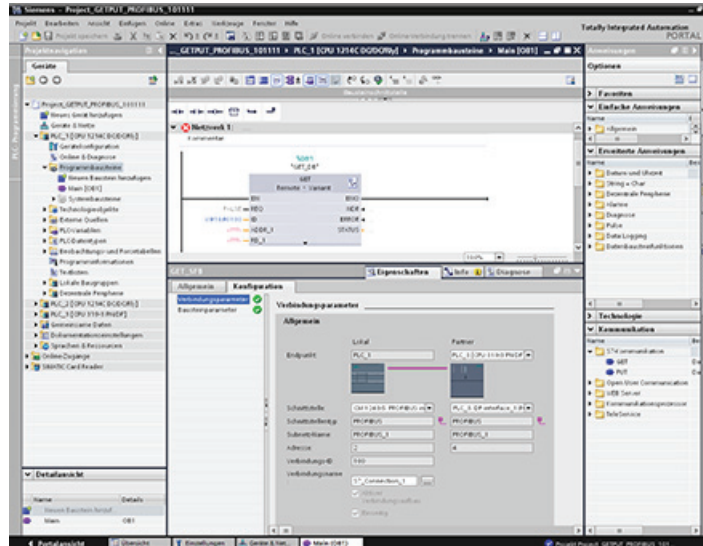
In der "Netzansicht" wird in der Tabelle "Verbindungen" zwischen "PLC\_1" und "PLC\_3" eine bidirektionale S7-Verbindung angezeigt.





## PROFIBUS S7-Verbindung konfigurieren

Wählen Sie für den "Partner-Endpunkt" die Option "PLC\_3" aus.



Das System reagiert mit den folgenden Änderungen:

Tabelle 10- 47 Verbindungsparameter: Allgemeine Werte

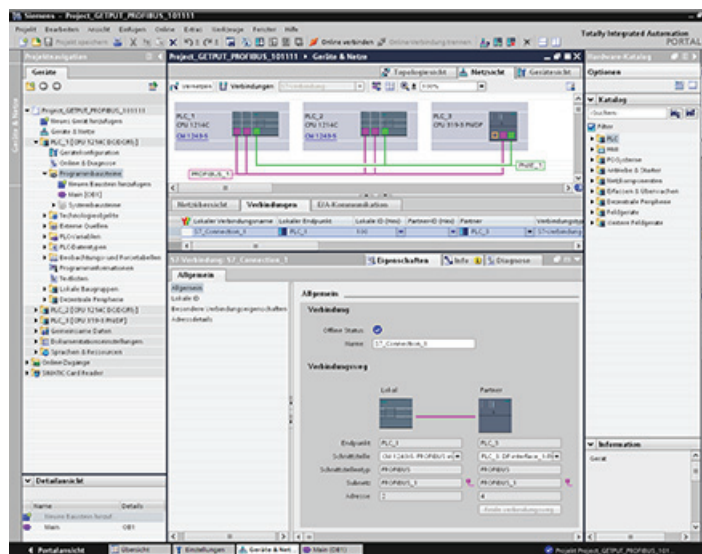
Parameter	Definition
Verbindungsparameter: Allgemein	Endpunkt "Lokaler Endpunkt" enthält "PLC_1" (schreibgeschützt). Das Feld "Partner-Endpunkt" enthält "PLC_3[CPU319-3PN/DP]": <ul style="list-style-type: none"> <li>Die Farbe wechselt von Rot nach Weiß.</li> <li>Das "Partner"-Gerätebild wird angezeigt.</li> <li>Eine Verbindungslinie erscheint zwischen den PLC_1- und PLC_3-Gerätebildern (violettfarbene PROFIBUS-Linie).</li> </ul>
	Schnittstelle "Lokale Schnittstelle" enthält "CPU1214C DC/DC/DC, PROFIBUS-Schnittstelle (R0/S1)". "Partner-Schnittstelle" enthält: "CPU319-3PN/DP, PROFIBUS-Schnittstelle (R0/S2)".
	Schnittstellentyp "Lokaler Schnittstellentyp" enthält "PROFIBUS"; die Bedienung ist schreibgeschützt. "Partner-Schnittstellentyp" enthält "PROFIBUS"; die Bedienung ist schreibgeschützt. Die Schnittstellentypbilder werden rechts neben dem lokalen und dem Partner-Schnittstellentyp angezeigt (violettfarbenedes PROFIBUS-Symbol).
	Subnetzname "Lokaler Subnetzname" enthält "PROFIBUS_1"; die Bedienung ist schreibgeschützt. "Partner-Subnetzname" enthält "PROFIBUS_1"; die Bedienung ist schreibgeschützt.

Parameter		Definition
	Adresse	"Lokale Adresse" enthält die lokale IP-Adresse; die Bedienung ist schreibgeschützt. "Partneradresse" enthält die Partner-IP-Adresse; die Bedienung ist schreibgeschützt.
	Verbindungs-ID	"Verbindungs-ID" enthält "100". Im Hauptprogramm [OB1] des Programmiereditors enthält die "Verbindungs-ID" des GET/PUT-Bausteins ebenfalls den Wert "100".
	Verbindungsname	"Verbindungsname" enthält den Standardverbindungsnamen (Beispiel: "S7_Verbindung_1"); die Bedienung ist aktiviert.
	Aktiver Verbindungsaufbau	Schreibgeschützt und aktiviert, um die lokale CPU als aktive Verbindung auszuwählen.
	Unidirektional	Schreibgeschützt und aktiviert. Hinweis: "PLC_3" (eine S7-300 CPU319-3PN/DP) kann in einer GET/PUT-Verbindung mit PROFIBUS nur als Server fungieren (und nicht auch als Client), so dass nur eine unidirektionale Verbindung möglich ist.

Das GET/PUT-Symbol in der Eigenschaftsansicht wechselt von Rot nach Grün.

### Vollständige PROFIBUS S7-Verbindung

In der "Netzsicht" wird in der Tabelle "Verbindungen" zwischen "PLC\_1" und "PLC\_3" eine unidirektionale S7-Verbindung angezeigt.



## Webserver

Der Webserver der S7-1200 bietet Webseitenzugriff auf Daten über Ihre CPU und auf Prozessdaten in der CPU.

### Standard-Webseiten

Die S7-1200 umfasst Standard-Webseiten, auf die Sie über einen Webbrowser auf Ihrem PC zugreifen (Seite 533) können:

- Einführung (Seite 537) - Einstiegspunkt in die Standard-Webseiten
- Startseite (Seite 538) - allgemeine Informationen über die CPU
- Identifikation (Seite 539) - ausführliche Informationen über die CPU einschließlich Serien-, Bestell- und Versionsnummer
- Modulinformationen (Seite 540) - Informationen zu den Modulen im lokalen Baugruppenträger
- Kommunikation (Seite 543) - Informationen über die Netzwerkadressen, physikalischen Eigenschaften der Kommunikationsschnittstellen und Kommunikationsstatistik
- Diagnosepuffer (Seite 540) - der Diagnosepuffer
- Variablenzustand (Seite 544) - CPU-Variablen und E/A, zugänglich über Adressen oder PLC-Variablennamen
- Datenprotokolle (Seite 546) - in der CPU oder auf einer Memory Card gespeicherte Datenprotokolle
- Firmware-Update (Seite 549) - Aktualisierung der Firmware in Ihrer CPU

Diese Seiten sind in die S7-1200 integriert. Einzelheiten zu den Standard-Webseiten und wie Sie darauf zugreifen finden Sie im Abschnitt Standard-Webseiten (Seite 533).

### Benutzerdefinierte Webseiten

Bei der S7-1200 können Sie auch benutzerdefinierte Webseiten erstellen, die auf CPU-Daten zugreifen können. Sie können diese Seiten mit einer HTML-Authoring-Software Ihrer Wahl anlegen und vordefinierte "AWP"-Befehl (Automation Web Programming) in Ihren HTML-Code einfügen, um auf CPU-Daten zuzugreifen. Im Kapitel Benutzerdefinierte Webseiten (Seite 551) finden Sie spezielle Informationen zum Entwickeln von benutzerdefinierten Webseiten und zur zugehörigen Konfiguration und Programmierung in STEP 7.

## Voraussetzungen für den Webbrowser

Die folgenden Webbrowser unterstützen den Webserver:

- Internet Explorer 8.0 oder höher
- Mozilla Firefox 3.0 oder höher
- Opera 11.0 oder höher


Informationen zu browserspezifischen Einschränkungen, die sich auf die Anzeige von Standard- oder Benutzer-Webseiten auswirken können, finden Sie im Abschnitt Einschränkungen (Seite 591).

## 11.1 Webserver aktivieren

Sie aktivieren den Webserver in STEP 7 über die Gerätekonfiguration der CPU, zu der Sie eine Verbindung herstellen möchten.

Um den Webserver zu aktivieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die CPU in der Gerätekonfiguration aus.
2. Wählen Sie im Inspektorfenster unter den CPU-Eigenschaften "Webserver" aus.
3. Aktivieren Sie das Kontrollkästchen für "Webserver auf diesem Modul aktivieren".
4. Um sicheren Zugriff auf die Standard-Webseiten zu verlangen, aktivieren Sie das Kontrollkästchen "Erlaubt nur HTTPS-Zugriff".

 <b>WARNUNG</b>
<p>Nicht berechtigter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.</p> <p>Weil durch die Aktivierung des Webserver "admin"-Benutzer Betriebszustandsänderungen vornehmen, PLC-Daten schreiben und Firmware-Updates durchführen können, empfiehlt Siemens, die folgenden Sicherheitsvorkehrungen einzuhalten:</p> <ul style="list-style-type: none"><li>• Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.</li><li>• Schützen Sie die CPU durch ein starkes Passwort (Seite 176). Starke Passwörter sind mindestens acht Zeichen lang, bestehen aus Buchstaben, Zahlen und Sonderzeichen, sind keine Wörter, die in einem Wörterbuch gefunden werden können, und sind keine Namen oder Kennungen, die sich aus persönlichen Daten ableiten lassen. Halten Sie das Passwort geheim und ändern Sie es häufig.</li><li>• Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil Nutzer von Webseiten für die PLC-Variablen ungültige Werte einrichten können.</li></ul>

Nachdem Sie die Gerätekonfiguration geladen haben, können Sie mit den Standard-Webseiten auf die CPU zugreifen. Wenn Sie für "Automatische Aktualisierung" die Option "Aktivieren" wählen, werden die Standard-Webseiten alle zehn Sekunden aktualisiert.

Wenn Sie benutzerdefinierte Webseiten angelegt haben, können Sie über das Menü der Standard-Webseiten darauf zugreifen.

---

**Hinweis**

Wenn ein "Laden in RUN" (Seite 735) durchgeführt wird, werden Datenwerte auf Standard- und benutzerdefinierten Webseiten nicht aktualisiert und Sie können auch erst dann wieder Datenwerte schreiben, wenn der Ladevorgang abgeschlossen ist. Jeder Versuch, während des Ladevorgangs Datenwerte zu schreiben, wird abgebrochen.

---

## 11.2 Standard-Webseiten

### 11.2.1 Über den PC auf die Standard-Webseiten zugreifen

Um über einen PC auf die S7-1200 Standard-Webseiten zuzugreifen, gehen Sie folgendermaßen vor:

1. Vergewissern Sie sich, dass sich die S7-1200 und der PC in einem gemeinsamen Ethernet-Netzwerk befinden oder über ein herkömmliches Ethernet-Kabel direkt miteinander verbunden sind.
2. Öffnen Sie einen Webbrowser und geben Sie die URL "http://ww.xx.yy.zz" ein. Dabei entspricht "ww.xx.yy.zz" der IP-Adresse der S7-1200 CPU.

Der Webbrowser öffnet die Einführungsseite.

---

**Hinweis**

Wenn Ihr Internetzugang keine direkte Verbindung mit einer IP-Adresse gestattet, wenden Sie sich an Ihren IT-Administrator. Ihre Webumgebung oder Ihr Betriebssystem weist möglicherweise andere Einschränkungen (Seite 591) auf.

---

Alternativ können Sie in Ihrem Webbrowser die Adresse einer spezifischen Standard-Webseite eingeben. Hierfür geben Sie die URL im Format "http://ww.xx.yy.zz/<Seite>.html" ein. Dabei entspricht <Seite> einer der folgenden Standard-Webseiten:

- start (Seite 538) - allgemeine Informationen über die CPU
- Identifikation (Seite 539) - ausführliche Informationen über die CPU einschließlich Serien-, Bestell- und Versionsnummer
- module (Seite 540) - Informationen zu den Modulen im lokalen Baugruppenträger
- Kommunikation (Seite 543) - Informationen über die Netzwerkadressen, physikalischen Eigenschaften der Kommunikationsschnittstellen und Kommunikationsstatistik
- diagnostic (Seite 540) - der Diagnosepuffer
- variable (Seite 544) - CPU-Variablen und E/A, zugänglich über Adressen oder PLC-Variablenamen

- Datenprotokoll (Seite 546) - in der CPU oder auf einer Memory Card gespeicherte Datenprotokolle
- updatefirmware (Seite 549) - Seite, über die sich die Firmware Ihrer CPU anhand einer Datei aktualisieren lässt
- index (Seite 537) - Einführungsseite für den Einstieg in die Standard-Webseiten

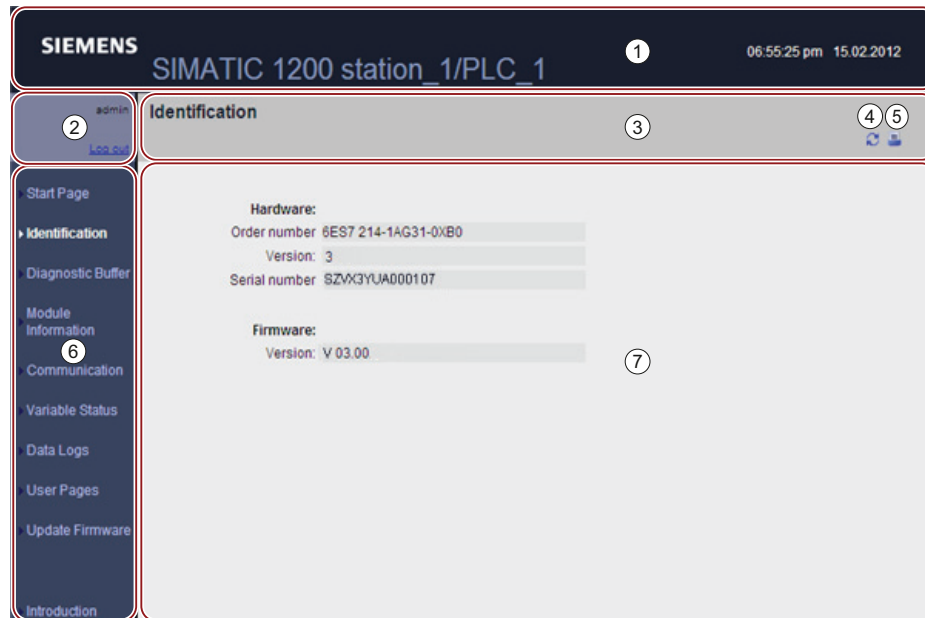
Wenn Sie beispielsweise "http://ww.xx.yy.zz/communication.html" eingeben, zeigt der Browser die Kommunikationsseite an.

### Sicherer Zugriff

Statt http:// können Sie für den sicheren Zugriff auf die Standard-Webseiten https:// verwenden. Wenn Sie über https:// eine Verbindung zur S7-1200 herstellen, verschlüsselt die Website die Sitzung mit einem digitalen Zertifikat. Die Daten werden sicher übertragen und sind nicht für jeden zugänglich. Üblicherweise wird eine Sicherheitswarnung angezeigt, die Sie mit "Ja" bestätigen, um die Standard-Webseiten aufzurufen. Um die Anzeige der Sicherheitswarnung bei jedem sicheren Zugriff zu vermeiden, können Sie das Siemens-Softwarezertifikat in Ihren Webbrowser importieren (Seite 593).

## 11.2.2 Aufbau der Standard-Webseiten

Alle Standard-Webseiten haben einen gemeinsamen Aufbau mit Links für die Navigation und Bedienelementen für die Seite. Diesen Aufbau sehen Sie im Folgenden:



- ① Kopfzeile im Webserver
- ② Anmelden oder Abmelden
- ③ Kopfzeile der Standard-Webseite mit dem Namen der angezeigten Seite. In diesem Beispiel handelt es sich um die Seite der CPU-Identifikation. Auf einigen der Standard-Webseiten, z. B. der Modulinformationsseite, wird hier auch ein Navigationspfad angezeigt, wenn mehrere Seiten dieses Typs aufrufbar sind.
- ④ Symbol zum Aktualisieren: bei Seiten mit automatischer Aktualisierung wird die automatische Aktualisierungsfunktion aktiviert bzw. deaktiviert; bei Seiten ohne automatische Aktualisierung wird die Seite mit den aktuellen Daten aktualisiert
- ⑤ Symbol zum Drucken: erstellt eine druckbare Version der Informationen auf der angezeigten Seite und zeigt sie an
- ⑥ Navigationsbereich zum Wechseln zu einer anderen Seite
- ⑦ Inhaltsbereich der spezifischen Standard-Webseite, die Sie gerade anzeigen. In diesem Beispiel handelt es sich um die Seite der CPU-Identifikation.

### Hinweis

#### Standard-Webseiten drucken

Beachten Sie beim Drucken von Standard-Webseiten, dass sich der gedruckte Inhalt gelegentlich von der angezeigten Seite unterscheiden kann. Der Ausdruck einer Diagnosepufferseite beispielsweise enthält möglicherweise neue Diagnoseeinträge, die nicht auf der Diagnosepufferseite angezeigt werden. Wenn die automatische Aktualisierung nicht aktiviert ist, zeigt die Seite die Diagnoseereignisse an, die zum Zeitpunkt des Seitenaufrufs vorlagen, während der Ausdruck die Diagnoseereignisse enthält, die zu dem Zeitpunkt vorlagen, als der Ausdruck angefordert wurde.

## Anmelden

Um die Daten auf den Standard-Webseiten anzuzeigen, ist keine Anmeldung erforderlich. Wenn Sie bestimmte Aktionen durchführen möchten, z. B. die Steuerung in einen anderen Betriebszustand versetzen, Werte in den Speicher schreiben oder die Firmware der CPU aktualisieren, müssen Sie sich als Benutzer "admin" anmelden.




The image shows a small rectangular login form with a light blue background. It contains two input fields: the top one is labeled 'Name' and the bottom one is labeled 'Password'. Below the password field is a blue button with the text 'Log in' in white.

Das Feld für die Anmeldung befindet sich oben links auf jeder Seite.

Um sich als Benutzer "admin" anzumelden, gehen Sie wie folgt vor:

1. Geben Sie als Namen "admin" ein.
2. Geben Sie ggf. das CPU-Passwort im Passwortfeld ein; andernfalls drücken Sie die Eingabetaste.

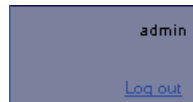
Nun sind Sie als Benutzer "admin" angemeldet.

 <b>WARNUNG</b>
<p>Nicht berechtigter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.</p> <p>Weil durch die Aktivierung des Webserver "admin"-Benutzer Betriebszustandsänderungen vornehmen, PLC-Daten schreiben und Firmware-Updates durchführen können, empfiehlt Siemens, die folgenden Sicherheitsvorkehrungen einzuhalten:</p> <ul style="list-style-type: none"> <li>• Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.</li> <li>• Schützen Sie die CPU durch ein starkes Passwort (Seite 176). Starke Passwörter sind mindestens acht Zeichen lang, bestehen aus Buchstaben, Zahlen und Sonderzeichen, sind keine Wörter, die in einem Wörterbuch gefunden werden können, und sind keine Namen oder Kennungen, die sich aus persönlichen Daten ableiten lassen. Halten Sie das Passwort geheim und ändern Sie es häufig.</li> <li>• Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil Nutzer von Webseiten für die PLC-Variablen ungültige Werte einrichten können.</li> </ul>

Wenn bei der Anmeldung Fehler auftreten, kehren Sie zur Einführungsseite (Seite 537) zurück und laden das Siemens-Sicherheitszertifikat herunter (Seite 593). Danach können Sie sich fehlerfrei anmelden.



## Abmelden



Um den Benutzer "admin" abzumelden, klicken Sie einfach auf einer beliebigen Seite auf den Link "Log out".

Sie können auch unangemeldet weiterhin Standard-Webseiten aufrufen und anzeigen, Sie können jedoch keine Aktionen durchführen, die auf den Benutzer "admin" beschränkt sind. In den Beschreibungen der einzelnen Standard-Webseiten werden ggf. die Aktionen erläutert, für die Sie sich als "admin" anmelden müssen.

### 11.2.3 Einleitung

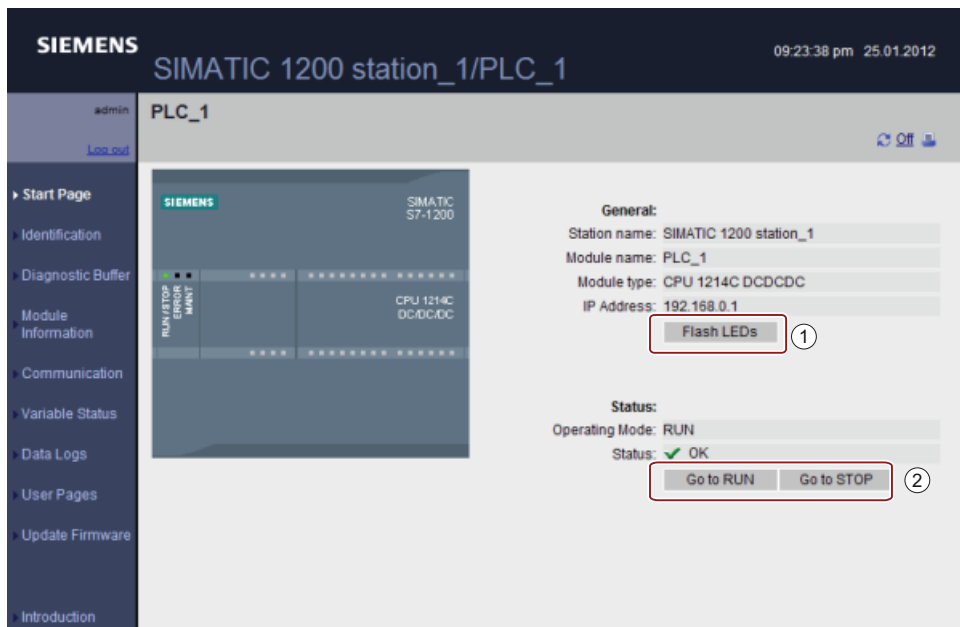
Die Einführungsseite ist die Begrüßungsseite für den Einstieg in die S7-1200 Standard-Webseiten.



Auf dieser Seite klicken Sie auf "Enter", um die S7-1200 Standard-Webseiten aufzurufen. Oben auf der Seite finden Sie Links zu nützlichen Websites von Siemens sowie einen Link, um das Siemens-Sicherheitszertifikat herunterzuladen (Seite 593).

### 11.2.4 Start

Die Startseite zeigt eine Darstellung der CPU an, mit der Sie verbunden sind, und führt allgemeine Informationen über die CPU auf. Wenn Sie sich als Benutzer "admin" anmelden, können Sie auch den Betriebszustand der CPU wechseln und die LEDs blinken lassen.

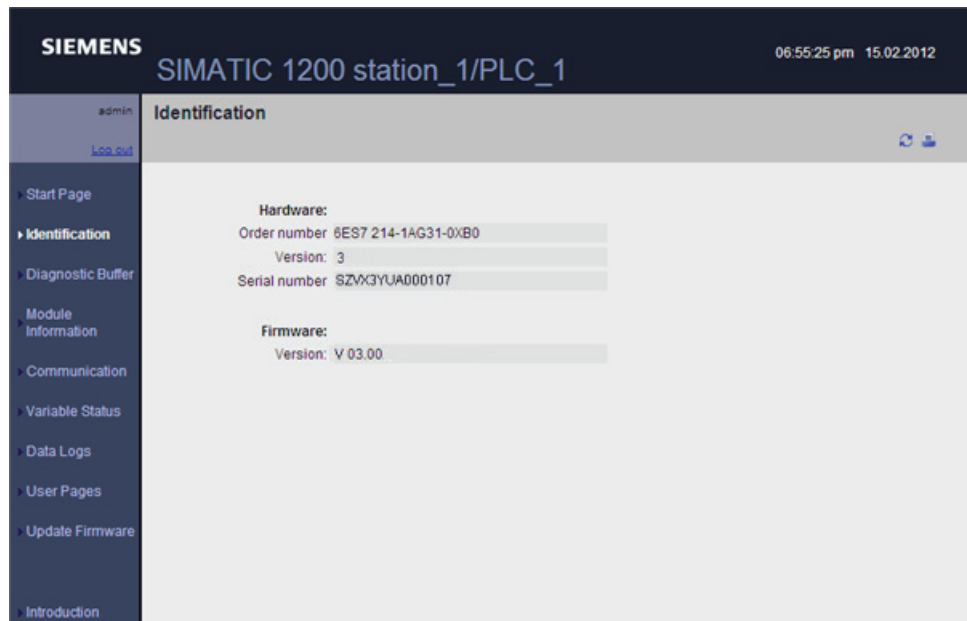


① und ② Die Schaltflächen zum Betätigen der LEDs und zum Wechseln des Betriebszustands werden nur dann auf der Startseite angezeigt, wenn Sie als "admin" angemeldet sind.

## 11.2.5 Identifikation

Die Identifikationsseite zeigt identifizierende Merkmale der CPU an:

- Seriennummer
- Bestellnummern
- Versionsinformationen

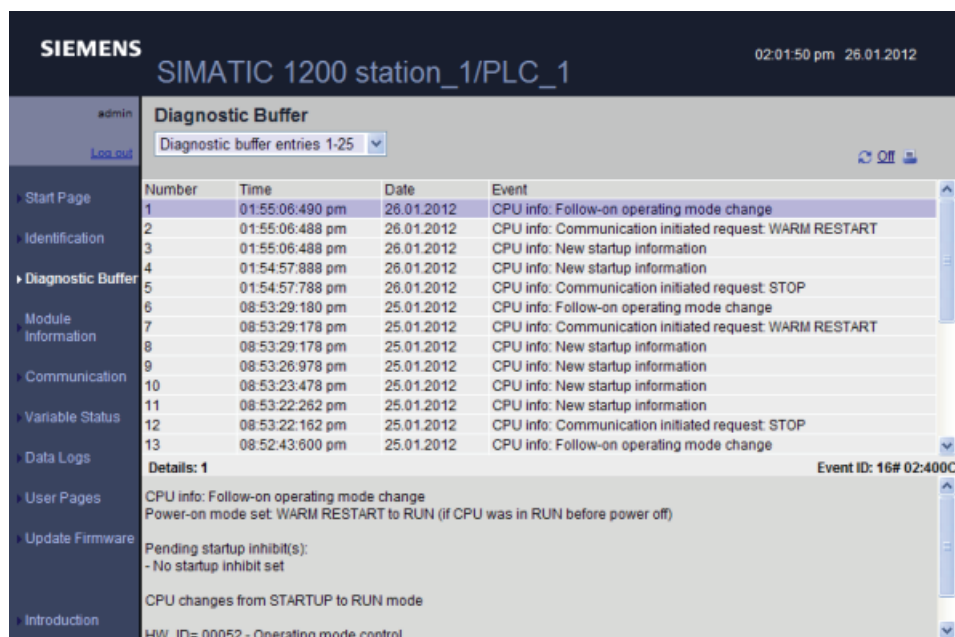


Die Identifikationsseite verändert sich nicht, wenn Sie sich als "admin" anmelden.

### 11.2.6 Diagnosepuffer

Die Diagnosepufferseite zeigt Diagnoseereignisse an. Sie können auswählen, welcher Bereich der Diagnosepuffereinträge angezeigt werden soll, entweder 1 bis 25 oder 26 bis 50. Im oberen Bereich der Seite werden diese Einträge mit Uhrzeit und Datum des Auftretens des Ereignisses in der CPU angezeigt. Bei den Uhrzeiten handelt es sich um die Systemzeit aus der Echtzeituhr (Seite 94) der CPU.

Im oberen Bereich der Seite können Sie einen einzelnen Eintrag auswählen, um dazu weitere Informationen im unteren Bereich der Seite anzuzeigen.



Die Diagnosepufferseite verändert sich nicht, wenn Sie sich als "admin" anmelden.

### 11.2.7 Modulinformationen

Die Modulinformationsseite zeigt Informationen zu allen Modulen im lokalen Baugruppenträger an. Im oberen Abschnitt der Seite wird ein Überblick über die Module angezeigt, im unteren Abschnitt werden Zustand und Identifikation des ausgewählten Moduls angezeigt.

## Statusanzeige

**SIEMENS** SIMATIC 1200 station\_1/PLC\_1 03:49:55 pm 26.01.2012

admin | [Log out](#)

**Module information** Slot

[SIMATIC 1200 station\\_1](#) - SIMATIC 1200 station\_1

Slot	Status	Name	Order number	I address	Q address	Comment
1	✓	<a href="#">PLC_1</a> <input type="button" value="Details"/>	6ES7 214-1AG31-0XB0	---	---	
2	✓	<a href="#">DQ16 x 24VDC_1</a> <input type="button" value="Details"/>	6ES7 222-1BH30-0XB0	---	8	
3	✓	<a href="#">DI16/DQ16 x 24VDC_1</a> <input type="button" value="Details"/>	6ES7 223-1BL30-0XB0	12	12	

Start Page | Identification | Diagnostic Buffer | **Module Information** | Communication | Variable Status | Data Logs | User Pages | Update Firmware | Introduction

**Status** | **Identification**

The module is parameterized, and it exchanges data with the CPU. The module did not detect any parameter assignment errors, channel errors, or hardware related errors.

## Identifikationsanzeige

**SIEMENS** SIMATIC 1200 station\_1/PLC\_1 03:51:25 pm 26.01.2012

admin | [Log out](#)

**Module information** Slot

[SIMATIC 1200 station\\_1](#) - SIMATIC 1200 station\_1

Slot	Status	Name	Order number	I address	Q address	Comment
1	✓	<a href="#">PLC_1</a> <input type="button" value="Details"/>	6ES7 214-1AG31-0XB0	---	---	
2	✓	<a href="#">DQ16 x 24VDC_1</a> <input type="button" value="Details"/>	6ES7 222-1BH30-0XB0	---	8	
3	✓	<a href="#">DI16/DQ16 x 24VDC_1</a> <input type="button" value="Details"/>	6ES7 223-1BL30-0XB0	12	12	

Start Page | Identification | Diagnostic Buffer | **Module Information** | Communication | Variable Status | Data Logs | User Pages | Update Firmware | Introduction

**Status** | **Identification**

Hardware Version: 1  
 Firmware Version: R.3.0.0  
 Serial Number: SZVX3YUA000107

### Weitere Einzelheiten anzeigen

Sie können im oberen Abschnitt einen Link auswählen, um weitere Einzelheiten der Modulinformationen des spezifischen Moduls anzuzeigen. Module mit Untermodulen haben für jedes Untermodul einen eigenen Link. Die Art der angezeigten Informationen richtet sich nach dem jeweils ausgewählten Modul. Zum Beispiel werden anfänglich in den Modulinformationen der Name der SIMATIC 1200 Station, eine Statusanzeige und ein Kommentar angezeigt. Wenn Sie den Link bis zur CPU-Ebene erweitern, zeigen die Modulinformationen die Namen der digitalen und analogen Eingänge und Ausgänge des CPU-Modells an (z. B. "DI14/DO10", "AI2") sowie Adressierungsinformationen für die E/A, Statusanzeigen, Steckplatznummern und Kommentare.

Slot	Status	Name	Order number	I address	Q address	Comment
1.16	✓	HSC_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1000	---	
1.17	✓	HSC_2	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1004	---	
1.18	✓	HSC_3	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1008	---	
1.19	✓	HSC_4	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1012	---	
1.20	✓	HSC_5	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1016	---	
1.21	✓	HSC_6	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1020	---	
1.2	✓	AI2_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	64	---	
1.1	✓	DI14/DO10_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	0	0	
1.32	✓	Pulse_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	1000	
1.33	✓	Pulse_2	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	1002	

Beim Erweitern der Anzeige gibt die Modulinformationsseite den entsprechenden Pfad an. Sie können auf jeden Link in diesem Pfad klicken, um zu einer höheren Ebene zurückzukehren.



### Felder zum Sortieren

Wenn in der Liste mehrere Module angezeigt werden, können Sie auf den Spaltentitel eines Felds klicken, um anhand dieses Felds auf- oder absteigend zu sortieren.

Name ▼	Details
AI2_1	<a href="#">Details</a>
DI14/DO10_1	<a href="#">Details</a>
HSC_1	<a href="#">Details</a>
HSC_2	<a href="#">Details</a>
HSC_3	<a href="#">Details</a>
HSC_4	<a href="#">Details</a>
HSC_5	<a href="#">Details</a>
HSC_6	<a href="#">Details</a>
Pulse_1	<a href="#">Details</a>
Pulse_2	<a href="#">Details</a>

### Modulinformationen filtern

Sie können jedes Feld in der Liste der Modulinformationen filtern. Wählen Sie in der Klappliste den Namen des Felds, dessen Daten Sie filtern möchten. Geben Sie Text in das zugehörige Textfeld ein und klicken Sie auf den Link "Filter". Die Liste wird aktualisiert und zeigt nur die Module an, die Ihren Filterkriterien entsprechen.

## Statusinformationen

Das Register "Status" im unteren Abschnitt der Modulinformationsseite zeigt eine Beschreibung des aktuellen Zustands des Moduls an, das im oberen Abschnitt ausgewählt ist.

## Identifikation

Das Register "Identifikation" zeigt die Seriennummer und die Versionsnummer des ausgewählten Moduls an.

Die Modulinformationsseite verändert sich nicht, wenn Sie sich als "admin" anmelden.

## 11.2.8 Kommunikation

Die Kommunikationsseite zeigt die Parameter der angeschlossenen CPU und die Kommunikationsstatistik an. Das Register "Parameter" zeigt die MAC-Adresse der CPU, die IP-Adresse und die IP-Einstellungen der CPU und die physikalischen Eigenschaften an. Das Register "Statistik" zeigt eine Statistik der Sende- und Empfangskommunikation an.

### Kommunikation: Parameteranzeige

The screenshot displays the Siemens SIMATIC 1200 web interface. The title bar shows 'SIEMENS SIMATIC 1200 station\_1/PLC\_1' and the time '04:02:50 pm 26.01.2012'. The user is logged in as 'admin'. The main content area is titled 'Communication' and has two tabs: 'Parameter' (selected) and 'Statistics'. The 'Parameter' tab shows the following information:

- Network connection:**
  - MAC address: 08-00-06-05-91-11
  - Name: plcxb1d0ed
- IP parameter:**
  - IP Address: 192.168.0.1
  - Subnet mask: 255.255.255.0
  - Default router: 0.0.0.0
  - IP settings: Retained IP address is set inside of project
- Physical properties:**

Port number	Link status	Settings	Mode
1	OK	automatic	100 MBit/s full-duplex

The left sidebar contains navigation links: Start Page, Identification, Diagnostic Buffer, Module Information, Communication (selected), Variable Status, Data Logs, User Pages, Update Firmware, and Introduction.

Kommunikation: Statistikanzeige



Die Kommunikationsseite verändert sich nicht, wenn Sie sich als "admin" anmelden.

11.2.9 Variablenzustand

Auf der Variablenzustandsseite können Sie alle E/A oder Speicherdaten in Ihrer CPU anzeigen. Sie können eine direkte Adresse (wie E0.0), einen PLC-Variablenamen oder eine Variable aus einem spezifischen Datenbaustein eingeben. Bei Datenbausteinvariablen setzen Sie den Bausteinnamen in doppelte Anführungszeichen. Für jeden überwachten Wert können Sie ein Anzeigeformat für die Daten auswählen. Sie können so lange im Rahmen der Seiteneinschränkung Werte eingeben, bis Sie alle gewünschten Werte spezifiziert haben. Die überwachten Werte werden automatisch angezeigt und standardmäßig aktualisiert, sofern Sie diese Aktualisierung nicht über "Off" oben rechts auf der Seite ausschalten. Ist die automatische Aktualisierung deaktiviert, können Sie sie über "On" wieder aktivieren.

Wenn Sie als "admin" angemeldet sind, können Sie auch Datenwerte ändern. Geben im entsprechenden Feld "Modify Value" die Werte ein, die Sie festlegen möchten. Klicken Sie neben einem Wert auf die Schaltfläche "Go", um den Wert in die CPU zu schreiben. Sie können auch mehrere Werte eingeben und auf "Modify All Values" klicken, um alle Werte in die CPU zu schreiben.



The screenshot shows the 'Variable Status' page for a SIMATIC 1200 station. The page title is 'SIMATIC 1200 station\_1/PLC\_1' and the user is logged in as 'admin'. The page contains a table of variables with the following columns: Address, Display Format, Monitor Value, Modify Value, and Modify. The 'Modify Value' column is highlighted with a red box, and a circled '1' is next to it. The 'Modify' column contains 'Go' buttons. Below the table are 'Monitor Value' and 'Modify All Values' buttons.

Address	Display Format	Monitor Value	Modify Value	Modify
Q0.1	BOOL	true	<input type="text"/>	Go
I0.1	BOOL	false	<input type="text"/>	Go
Conveyor_speed	DEC	145	<input type="text"/>	Go
Mixer_on	BOOL	false	<input type="text"/>	Go
Start_Mixer	BOOL	false	<input type="text"/>	Go
MW10	DEC	12	<input type="text"/>	Go
"Data_block_1".flag1	BOOL	2#1	<input type="text"/>	Go
"Data_block_1".location	STRING	*	<input type="text"/>	Go
Tag_1	FLOATING_POINT	0.0	<input type="text"/>	Go
New variable	BOOL		<input type="text"/>	

① Die Funktionalität "Modify Value" ist nur sichtbar und zugänglich, wenn Sie als Benutzer "admin" angemeldet sind.

Wenn Sie die Variablenzustandsseite verlassen und wieder zurückkehren, sind Ihre Eingaben auf der Variablenzustandsseite nicht gespeichert. Sie können die Seite als Lesezeichen speichern und das Lesezeichen erneut aufrufen, damit die gleichen Einträge angezeigt werden. Wenn Sie die Seite nicht als Lesezeichen speichern, müssen Sie die Variablen erneut eingeben.

### Hinweis

Beachten Sie bei der Nutzung der Standardseite für den Variablenzustand Folgendes:

- Auf der Variablenzustandsseite können Sie keine Zeichenketten mit mehr als 198 Zeichen ändern.
- Gehen Sie bei der Eingabe eines Werts vom Datentyp Real oder LReal in Exponentialdarstellung auf der Variablenzustandsseite wie folgt vor:
  - Um einen Realzahlenwert (Real oder LReal) mit positivem Exponenten (wie +3.402823e+25) einzugeben, geben Sie den Wert in einem der folgenden Formate ein:
    - +3.402823e25
    - +3.402823e+25
  - Um einen Realzahlenwert (Real oder LReal) mit negativem Exponenten (wie +3.402823e-25) einzugeben, geben Sie den Wert wie folgt ein:
    - +3.402823e-25
  - Achten Sie darauf, dass der Anteil der Mantisse des Realzahlenwerts in der Exponentialdarstellung einen Dezimalpunkt enthält. Wird der Dezimalpunkt weggelassen, führt dies dazu, dass der Wert zu einem unerwarteten ganzzahligen Wert verändert wird. Beispiel: Geben Sie -1.0e8 und nicht -1e8 ein.
- Die Variablenzustandsseite unterstützt nur 15 Ziffern für einen Wert vom Typ LReal (unabhängig von der Position des Dezimalpunkts). Wenn Sie mehr als 15 Ziffern eingeben, entsteht ein Rundungsfehler.

Einschränkungen auf der Variablenzustandsseite:

- Die maximale Anzahl von Variableneinträgen pro Seite beträgt 50.
- Die maximale Anzahl von Zeichen für die URL der Variablenzustandsseite beträgt 2083. Sie sehen die URL, die Ihrer aktuellen Variablenseite entspricht, in der Adressleiste Ihres Browsers.
- Hinsichtlich des Zeichenanzeigeformats zeigt die Seite Hexadezimalwerte an, wenn es sich nach Auswertung des Browsers bei den tatsächlichen CPU-Werten nicht um gültige ASCII-Zeichen handelt.

---

#### **Hinweis**

Wenn ein Variablenname Sonderzeichen enthält, so dass der Name als Eintrag auf der Variablenzustandsseite abgelehnt wird, können Sie den Variablennamen in doppelte Anführungszeichen setzen. In den meisten Fällen erkennt dann die Variablenzustandsseite den Variablennamen.

---

### **11.2.10 Datenprotokolle**

Auf der Datenprotokollseite können Sie eine angegebene Anzahl von Datenprotokolleinträgen anzeigen oder herunterladen. Wenn Sie als "admin" angemeldet sind, können Sie die Einträge nach dem Herunterladen auch zurücksetzen oder löschen. Der Webserver lädt die Datenprotokolle im CSV-Format (durch Komma getrennte Werte) auf Ihren PC.

Auf der Datenprotokollseite werden bis zu 40 Datenprotokolldateien angezeigt. Wenn mehr als 40 Datenprotokolldateien vorhanden sind, zeigt die Datenprotokollseite die ersten 40 Dateien an.

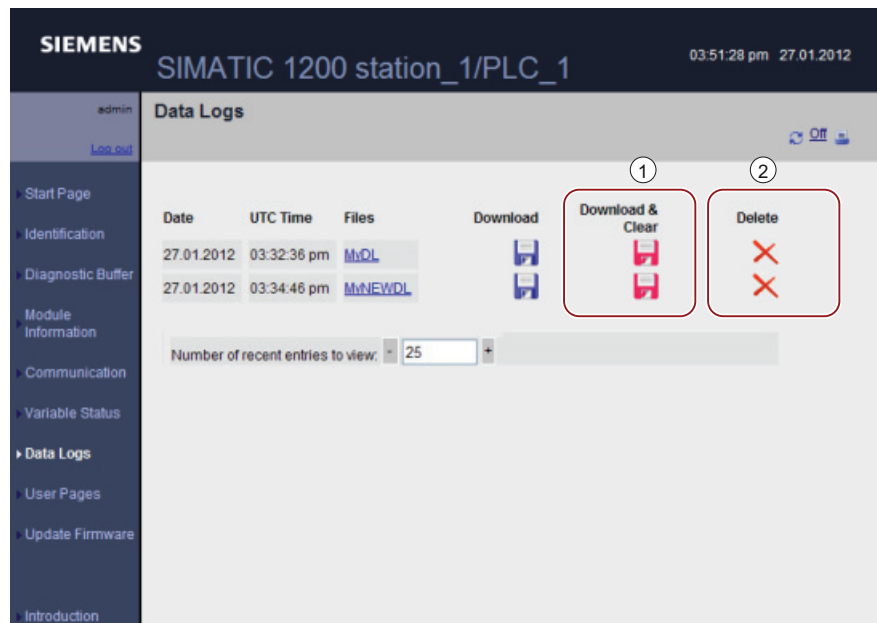
---

#### **Hinweis**

##### **Die Zeitstempel der Datenprotokolle zeigen die Systemzeit, nicht die Lokalzeit**

Die CPU gibt den Zeitstempel der Datenprotokolleinträge in der Systemzeit (Seite 94) an, und auch die Standardseite "Datenprotokolle" des Webserver zeigt die Zeitstempel der Datenprotokolle in Systemzeit an.

---



- ① Die Option "Download & Clear" ist nur verfügbar, wenn Sie als Benutzer "admin" angemeldet sind.
- ② Die Option "Delete" ist nur verfügbar, wenn Sie als Benutzer "admin" angemeldet sind.

### Hinweis

Die Protokolldatei steht im amerikanischen/englischen CSV-Format (durch Komma getrennte Werte). Um dieses Format in Excel auf nicht amerikanischen/englischen Systemen zu öffnen, müssen Sie die Datei mit spezifischen Einstellungen in Excel importieren (Seite 595).

### Recent entries: Herunterladen einer angegebenen Anzahl zuletzt gespeicherter Dateneinträge

Legen Sie die Anzahl zuletzt gespeicherter Einträge zum Herunterladen fest, und klicken Sie dann auf den Namen des Datenprotokolls, um den Ladevorgang der angegebenen Anzahl von Einträgen zu beginnen. In der CSV-Ausgabedatei sind die Dateneinträge in absteigender Reihenfolge der Einträge sortiert. Sie werden von Windows aufgefordert, die Protokolldatei zu öffnen oder zu speichern.

Standardmäßig beträgt die angezeigte Anzahl zuletzt gespeicherter Einträge 25. Sie können diesen Wert durch Eingabe einer Zahl in das Feld "Number of recent entries to view" ändern, oder indem Sie mit den Schaltflächen + und - den Wert vergrößern bzw. verkleinern.

### Hinweis

Der Parameter Records der Anweisung DataLogCreate (Seite 335) legt die maximale Anzahl von Einträgen je Datenprotokolldatei fest.

### Herunterladen einer Protokolldatei mit allen Dateneinträgen

Um eine komplette Protokolldatei herunterzuladen, klicken Sie auf den Download-Link der gewünschten Protokolldatei. Sie werden von Windows aufgefordert, die Protokolldatei zu öffnen oder zu speichern.

Die CSV-Ausgabedatei enthält alle Einträge. Diese sind in absteigender Reihenfolge nach der Eintragsnummer sortiert, sofern das Datenprotokoll nicht voll ist und ältere Einträge (mit niedrigeren Eintragsnummern) von neueren Einträgen (mit höheren Eintragsnummern) überschrieben werden.

### Protokolldatei herunterladen und zurücksetzen

Um eine Protokolldatei herunterzuladen und dann alle Dateneinträge zurückzusetzen, müssen Sie als Benutzer "admin" angemeldet sein. Klicken Sie dann auf das entsprechende Symbol zum Herunterladen und Zurücksetzen der gewünschten Protokolldatei. Sie werden von Windows aufgefordert, die Protokolldatei zu öffnen oder zu speichern.

Nach Abschluss des Ladevorgangs wird in der im PLC-Gerät gespeicherten Datenprotokolldatei nach dem Header-Eintrag eine neue Zeile "//END" eingefügt. Dadurch wird das Datenprotokoll effektiv für die künftige interne PLC-Verarbeitung zurückgesetzt, doch bei nachfolgenden Ladevorgängen dieser Datei werden oberhalb der ersten Zeile "//END" neue Dateneinträge eingefügt.

---

#### Hinweis

##### **Kennzeichnung "//END" in der CSV-Datei des Datenprotokolls**

Die Abschlussmarkierung "//END" in der CSV-Datei wird nur bei den ersten ((max. Anzahl Einträge) -1) Einträgen verwendet, um das logische Ende der Datei zu kennzeichnen. Nach dem logischen Ende kann die Datei Daten enthalten, die von Excel als zusätzliche Dateneinträge interpretiert werden können. Suchen Sie nach dem ersten "//END", löschen Sie diese Zeile und alle darunter aufgeführten Einträge. Ist keine logische Abschlusskennzeichnung vorhanden, können Sie die Datenzeilen nach der Eintragsnummer sortieren.

---

### Protokolldatei löschen

Um eine Protokolldatei zu löschen, müssen Sie als Benutzer "admin" angemeldet sein. Klicken Sie dann auf das entsprechende Symbol zum Löschen der gewünschten Protokolldatei. Der Webserver löscht dann die ausgewählte Protokolldatei.

### Weitere Informationen

Weitere Informationen zur Programmierung mit den Datenprotokollanweisungen finden Sie im Kapitel Datenprotokollierung (Seite 333).

## 11.2.11 Firmware aktualisieren

Auf der Seite für das Firmware-Update kann ein Benutzer "admin" die Firmware der CPU über eine Datei aktualisieren.

---

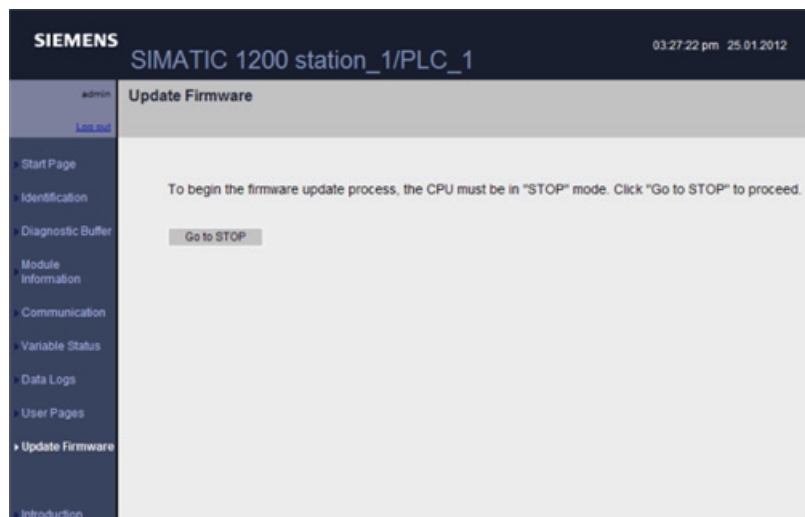
### Hinweis

Sie können nur S7-1200 CPUs ab Version 3.0 mit der Firmware-Update-Funktion aktualisieren.

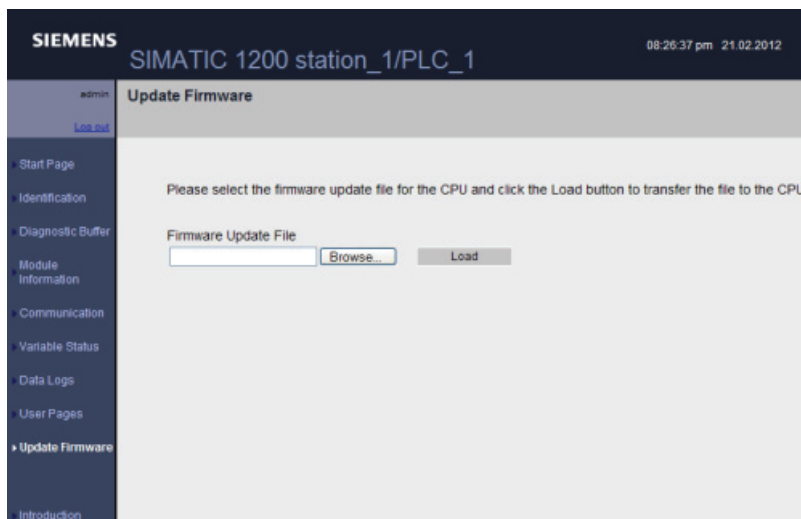
Der Webserver verwendet das "https"-Protokoll, um das Firmware-Update durchzuführen.

---

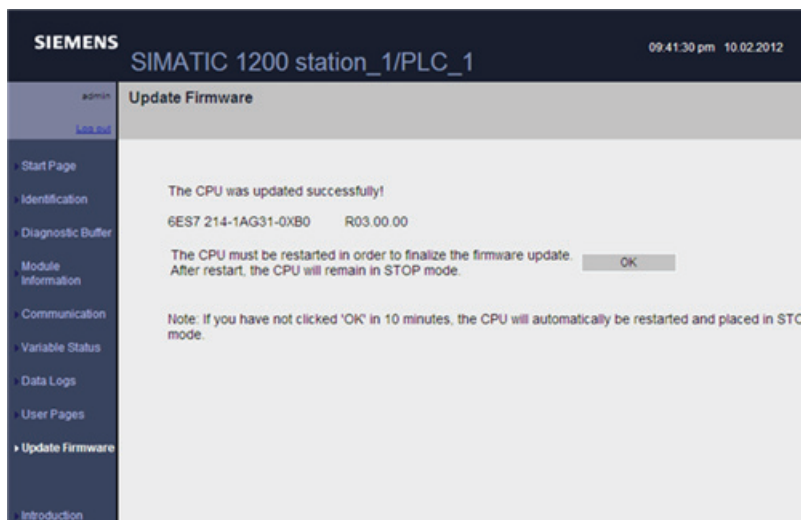
Die CPU muss sich im Betriebszustand STOP befinden, um ein Firmware-Update durchzuführen. Sollte sich die CPU noch nicht im Betriebszustand STOP befinden, werden Sie vom Webserver aufgefordert, die CPU in den Betriebszustand STOP zu versetzen.



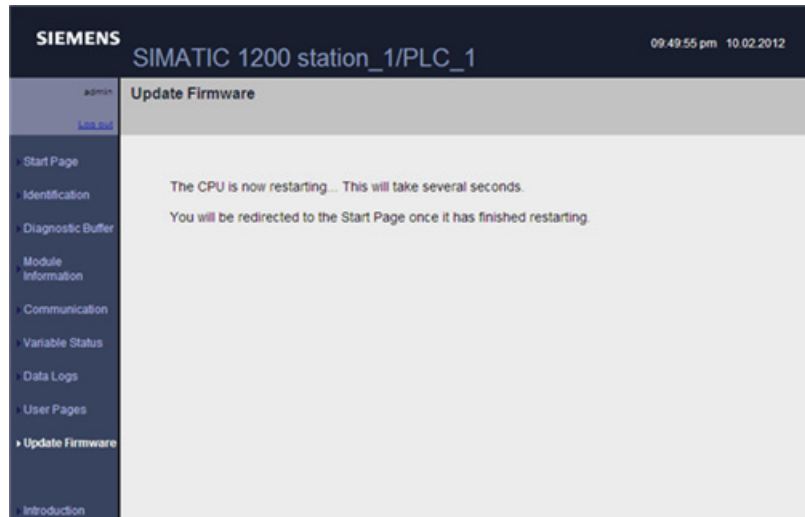
Wenn sich die CPU in STOP befindet, können Sie eine Datei auswählen, über die das Firmware-Update geladen werden soll. Firmware-Updates stehen auf der Kunden-Support-Website (<http://support.automation.siemens.com>) zur Verfügung. Nachdem Sie das entsprechende Firmware-Update von der Website auf Ihren Computer geladen haben, können Sie die Datei für die Aktualisierung auf Ihrem Computer auswählen.



Während der Aktualisierung wird auf der Seite für das Firmware-Update eine Meldung angezeigt, dass die Aktualisierung gerade durchgeführt wird. Nach Abschluss der Aktualisierung werden auf der Seite für das Firmware-Update die Bestellnummer und die Versionsnummer der aktualisierten Firmware angezeigt. Zudem werden Sie aufgefordert, einen Neustart der CPU zuzulassen.



Wenn Sie nicht innerhalb von zehn Minuten reagieren, wird die CPU automatisch neu gestartet.

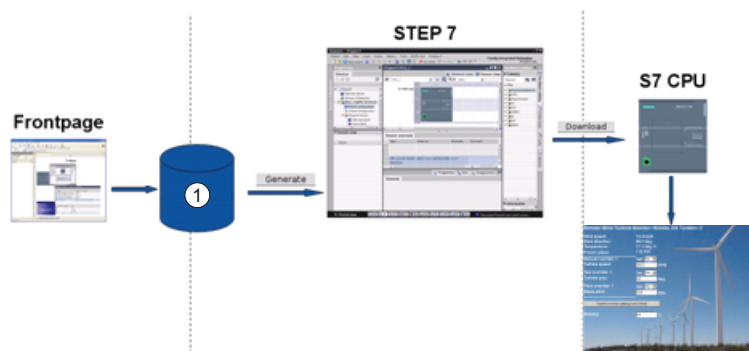


## 11.3 Benutzerdefinierte Webseiten

Für den S7-1200 Webserver können Sie auch eigene anwendungsspezifische HTML-Seiten mit Daten des Zielsystems anlegen. Sie erstellen diese Seiten mit einem HTML-Editor Ihrer Wahl und laden die Seiten in die CPU, von wo aus sie über das Menü der Standard-Webseiten aufrufbar sind. Hierfür sind mehrere Tätigkeiten durchzuführen:

- HTML-Seiten mit einem HTML-Editor wie Microsoft Frontpage anlegen (Seite 552)
- AWP-Befehle in HTML-Kommentare im HTML-Code einfügen (Seite 553): Bei den AWP-Befehlen handelt es sich um einen fest vorgegebenen Satz Befehle, den Siemens für den Zugriff auf CPU-Informationen bereitstellt.
- STEP 7 zum Lesen und Verarbeiten von HTML-Seiten konfigurieren (Seite 567)
- Bausteine aus den HTML-Seiten generieren (Seite 567)
- STEP 7 für die Steuerung der Verwendung der HTML-Seiten programmieren (Seite 569)
- Bausteine übersetzen und in die CPU laden (Seite 571)
- Über Ihren PC auf die benutzerdefinierten Webseiten zugreifen (Seite 571)

Dieser Prozess wird im Folgenden dargestellt:



① HTML-Dateien mit eingebetteten AWP-Befehlen

### 11.3.1 HTML-Seiten anlegen

Um Ihre eigenen HTML-Seiten zur Verwendung mit dem Webserver zu erstellen, können Sie ein Softwarepaket Ihrer Wahl verwenden. Achten Sie darauf, dass Ihr HTML-Code mit den HTML-Standards des W3C (World Wide Web Consortium) konform ist. STEP 7 führt keine Überprüfung Ihrer HTML-Syntax durch.

Sie können eine Software verwenden, bei der Sie im WYSIWYG- oder Design-Layout-Modus programmieren können, doch Sie müssen Ihren HTML-Code auch im reinen HTML-Format bearbeiten können. Die meisten Web-Authoring-Tools bieten diese Art der Bearbeitung. Falls nicht, können Sie den HTML-Code einfach in einem Texteditor bearbeiten. Nehmen Sie die folgende Zeile in Ihre HTML-Seite auf, um UTF-8 als Zeichensatz der Seite festzulegen:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Und achten Sie darauf, dass Sie die Datei im Editor ebenfalls in der UTF-8-Zeichencodierung speichern:

Sie können in STEP 7 die Inhalte Ihrer HTML-Seiten in STEP 7-Datenbausteine übersetzen. Diese Datenbausteine bestehen aus einem Steuerdatenbaustein, der die Anzeige der Webseiten regelt, und einem oder mehreren Datenbausteinfragmenten mit den übersetzten Webseiten. Beachten Sie, dass umfangreiche HTML-Seiten, insbesondere Seiten mit vielen Bildern, einen beträchtlichen Platz im Ladespeicher (Seite 571) für die DB-Fragmente belegen. Wenn der interne Ladespeicher Ihrer CPU für Ihre benutzerdefinierten Webseiten nicht ausreicht, stellen Sie über eine Memory Card (Seite 116) externen Ladespeicher zur Verfügung.

Um Ihren HTML-Code so zu programmieren, dass Daten aus der S7-1200 verwendet werden, können Sie AWP-Befehle (Seite 553) als HTML-Kommentare einfügen. Speichern Sie abschließend Ihre HTML-Seiten auf Ihrem PC und notieren Sie sich den Speicherpfad.



## Benutzerdefinierte Webseiten aktualisieren

Benutzerdefinierte Webseiten werden nicht automatisch aktualisiert. Sie können wählen, ob Sie den HTML-Code so programmieren, dass die Seite aktualisiert wird oder nicht. Bei Seiten, die PLC-Daten anzeigen, bleiben die Daten durch regelmäßige Aktualisierung auf dem aktuellen Stand. Bei HTML-Seiten, die als Formulare zur Dateneingabe dienen, kann eine Aktualisierung die Dateneingabe durch den Benutzer beeinträchtigen. Wenn Ihre gesamte Seite automatisch aktualisiert werden soll, können Sie diese Zeile in Ihre HTML-Kopfzeile aufnehmen, wobei "10" für die Anzahl der Sekunden zwischen zwei Aktualisierungsvorgängen steht:

```
<meta http-equiv="Refresh" content="10">
```

Sie können für die Seiten- und Datenaktualisierung auch JavaScripts oder andere HTML-Techniken nutzen. Hierfür ziehen Sie bitte eine Dokumentation zu HTML oder JavaScripts hinzu.

### 11.3.2 Vom S7-1200 Webserver unterstützte AWP-Befehle

Der S7-1200 Webserver bietet AWP-Befehle, die Sie zu folgenden Zwecken als HTML-Befehle in Ihre benutzerdefinierten Webseiten einfügen können:

- Variablen lesen (Seite 554)
- Variablen schreiben (Seite 555)
- Sondervariablen lesen (Seite 557)
- Sondervariablen schreiben (Seite 559)
- Enum-Typen definieren (Seite 561)
- Variablen zu Enum-Typen zuweisen (Seite 562)
- Datenbausteinfragmente erstellen (Seite 563)

#### Allgemeine Syntax

Mit Ausnahme des Befehls zum Lesen einer Variablen haben die AWP-Befehle die folgende Syntax:

```
<!-- AWP_ <Befehlsname und Parameter> -->
```

Sie nutzen die AWP-Befehle in Verbindung mit typischen HTML-Formularbefehlen, um Variablen in die CPU zu schreiben.

In den Beschreibungen der AWP-Befehle auf den folgenden Seiten werden die folgenden Konventionen verwendet:

- In eckigen Klammern [ ] angegebene Elemente sind optional.
- In spitzen Klammern < > dargestellte Elemente sind anzugebende Parameterwerte.
- Fragezeichen sind ein tatsächlicher Teil des Befehls. Sie müssen wie angezeigt vorhanden sein.
- Sonderzeichen in Variablen- oder Datenbausteinnamen müssen, je nach Verwendung, durch eine Escape-Zeichenfolge gekennzeichnet oder in Anführungszeichen gesetzt werden (Seite 565).

Fügen Sie die AWP-Befehle mit einem Texteditor oder im HTML-Bearbeitungsmodus in Ihre Webseiten ein.

## Überblick über die AWP-Befehle

Genauere Erläuterungen zu den einzelnen AWP-Befehlen finden Sie in den nächsten Abschnitten, hier zunächst ein Überblick über die Befehle:

**Variablen lesen**  
`:=<Varname>:`

**Variablen schreiben**

`<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->`

Dieser AWP-Befehl deklariert lediglich, dass in die Variable der Namensklauseel geschrieben werden kann. Ihr HTML-Code führt namentliche Schreibvorgänge in die Variable aus `<input>`, `<select>` oder einer anderen HTML-Anweisung innerhalb eines HTML-Formulars durch.

**Sondervariablen lesen**

`<!-- AWP_Out_Variable Name='<Typ>:<Name>' [Use='<Varname>'] -->`

**Sondervariablen schreiben**

`<!-- AWP_In_Variable Name='<Typ>:<Name>' [Use='<Varname>'] -->`

**Enum-Typen definieren**

`<!-- AWP_Enum_Def Name='<Name Enum-Typ>' Values='<Wert>, <Wert>, ... ' -->`

**Enum-Typen definieren**

`<!-- AWP_Enum_Ref Name='<Varname>' Enum='<Enum-Typ>' -->`

**Fragmente erstellen**

`<!-- AWP_Start_Fragment Name='<Name>' [Type=<Typ>] [ID=<ID>] -->`

**Fragmente importieren**

`<!-- AWP_Import_Fragment Name='<Name>' -->`

### 11.3.2.1 Variablen lesen

Benutzerdefinierte Webseiten können Variablen (PLC-Variablen) aus der CPU lesen.

#### Syntax

`:=<Varname>:`

## Parameter

<Varname>	Die zu lesende Variable, bei der es sich um eine PLC-Variable aus Ihrem STEP 7-Programm, eine Datenbausteinvariable, E/A oder eine Adresse im Speicher handeln kann. Setzen Sie bei Speicher- und E/A-Adressen oder Aliasnamen (Seite 565) den Variablennamen nicht in Anführungszeichen. Setzen Sie bei PLC-Variablen den Variablennamen in doppelte Anführungszeichen. Bei Datenbausteinvariablen setzen Sie nur den Bausteinnamen in doppelte Anführungszeichen. Der Variablenname befindet sich außerhalb der Anführungszeichen. Beachten Sie, dass Sie den Namen des Datenbausteins und nicht die Nummer des Datenbausteins verwenden.
-----------	---

## Beispiele

```

:="Fördergeschwindigkeit":="Mein_Datenbaustein".Merker1:
:=E0.0:
:=MW100:

```

### Beispiel für das Lesen einer über Alias angegebenen Variable

```

<!-- AWP_Out_Variable Name='Merker1'
Use=' "Mein_Datenbaustein".Merker1' -->
:=Merker1:

```

---

#### Hinweis

Wie Sie Aliasnamen für PLC-Variablen und Datenbausteinvariablen definieren, wird unter Alias für einen Variablenverweis nutzen (Seite 560) beschrieben.

---

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzlich Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablennamen mit Sonderzeichen (Seite 565) beschrieben.

### 11.3.2.2 Variablen schreiben

Benutzerdefinierte Webseiten können Daten in die CPU schreiben. Hierfür geben Sie über einen AWP-Befehl eine Variable in der CPU an, in die über die HTML-Seite geschrieben werden soll. Die Variable muss über den PLC-Variablennamen oder den Variablennamen des Datenbausteins angegeben werden. Sie können in einer Anweisung mehrere Variablennamen deklarieren. Um die Daten in die CPU zu schreiben, verwenden Sie den Standard-HTTP-Befehl POST.

Eine typische Verwendung ist die Gestaltung eines Formulars auf Ihrer HTML-Seite mit Texteingabefeldern oder Listenauswahlfeldern, die den schreibbaren CPU-Variablen entsprechen. Wie bei allen benutzerdefinierten Webseiten generieren Sie dann die Bausteine in STEP 7, damit diese in Ihr STEP 7-Programm aufgenommen werden. Wenn ein admin-Benutzer anschließend diese Seite aufruft und Daten in die Eingabefelder einträgt oder ein Feld in einer Auswahlliste auswählt, konvertiert der Webserver die Eingaben in den entsprechenden Datentyp für die Variablen und schreibt den Wert in die Variablen in der CPU. Beachten Sie, dass die Namensklausele für HTML-Eingabefelder und HTML-Auswahllisten eine Syntax verwendet, die für die Namensklausele des Befehls AWP\_In\_Variable typisch ist. Üblicherweise setzen Sie den Namen in einfache Anführungszeichen und beim Verweis auf einen Datenbaustein setzen Sie den Datenbausteinnamen in doppelte Anführungszeichen.

Weitere Informationen zur Formularverwaltung finden Sie in der HTML-Dokumentation.

**Syntax**

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

**Parameter**

<Varname1>	Wenn keine Use-Klausele angegeben ist, ist Varname1 die Variable, in die geschrieben wird. Hierbei kann es sich um eine PLC-Variable aus Ihrem STEP 7-Programm oder um eine Variable aus einem spezifischen Datenbaustein handeln.  Ist eine Use-Klausele angegeben, ist Varname1 ein alternativer Name für die in <Varname2> referenzierte Variable (Seite 560). Es handelt sich um einen lokalen Namen innerhalb der HTML-Seite.
<Varname2>	Wenn eine Use-Klausele angegeben ist, ist Varname2 die Variable, in die geschrieben wird. Hierbei kann es sich um eine PLC-Variable aus Ihrem STEP 7-Programm oder um eine Variable aus einem spezifischen Datenbaustein handeln.

Sowohl bei Namens- als auch bei Verwendungsklauseln ist der vollständige Name in einfache Anführungszeichen zu setzen. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinnamen in doppelte Anführungszeichen. Der Datenbausteinname befindet sich in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden.

**Beispiele mit HTML-Eingabefeldern**

```
<!-- AWP_In_Variable Name='\"Target_Level\"' -->
<form method="post">
<p>Input Target Level: <input name='\"Target_Level\"' type="text" />
</p>
</form>
```

```
<!-- AWP_In_Variable Name='\"Data_block_1\".Braking' -->
<form method="post">
<p>Braking: <input name='\"Data_block_1\".Braking' type="text" />
%</p>
</form>
```

```

<!-- AWP_In_Variable Name='Braking' Use='Data_block_1.Braking' -
->
<form method="post">
<p>Braking: <input name="Braking" type="text" /> %</p>
</form>

```

### Beispiel mit einer HTML-Auswahlliste

```

<!-- AWP_In_Variable Name='Data_block_1.ManualOverrideEnable'-->
<form method="post">
<select name='Data_block_1.ManualOverrideEnable'>
<option value="Data_block_1.ManualOverrideEnable:> </option>
<option value=1>Yes</option>
<option value=0>No</option>
</select><input type="submit" value="Submit setting" /></form>

```

---

#### Hinweis

Nur der admin-Benutzer kann Daten in die CPU schreiben. Die Befehle werden ignoriert, wenn sich der Benutzer nicht als admin angemeldet hat.

---

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter "Handhabung von Variablennamen mit Sonderzeichen (Seite 565)" beschrieben.

### 11.3.2.3 Sondervariablen lesen

Der Webserver bietet die Möglichkeit, Werte aus dem PLC-Gerät zu lesen und diese in Sondervariablen in der HTTP-Antwortkopfzeile zu speichern. Sie können z. B. einen Pfadnamen aus einer PLC-Variable auslesen, um die URL über die Sondervariable HEADER:Speicherort zu einem anderen Speicherort umzuleiten.

#### Syntax

```

<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->

```

**Parameter**

<Typ>	Der Typ der Sondervariable. Möglich sind: HEADER COOKIE_VALUE COOKIE_EXPIRES
<Name>	In der HTTP-Dokumentation finden Sie eine Liste aller Namen der HEADER-Variablen. Hier werden einige Beispiele aufgeführt: Status: Antwortcode Location: Pfad für die Umleitung Retry-After: Zeitdauer, über die der Dienst dem anfordernden Client voraussichtlich nicht zur Verfügung steht. Bei den Typen COOKIE_VALUE und COOKIE_EXPIRES, ist <Name> der Name eines bestimmten Cookies. COOKIE_VALUE:name: Wert des genannten Cookies COOKIE_EXPIRES:name: Ablaufzeit in Sekunden des genannten Cookies Die Namensklausele muss in einfache oder doppelte Anführungszeichen gesetzt werden. Ist keine Verwendungsklausel angegeben, entspricht der Sondervariablenname einem PLC-Variablennamen. Setzen Sie die vollständige Namensklausele in einfache Anführungszeichen und die PLC-Variable in doppelte Anführungszeichen. Der Sondervariablenname und der PLC-Variablenname müssen sich exakt entsprechen.
<Varname>	Name der PLC-Variable oder der Datenbausteinvariable für die auszulesende Variable Der Variablenname muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinnamen in doppelte Anführungszeichen. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden.

**Beispiele**

```
<!-- AWP_Out_Variable Name=' "HEADER:Status" ' -->
```

In diesem Beispiel empfängt die HTTP-Sondervariable "HEADER:Status" den Wert der PLC-Variable "HEADER:Status". Der Name in der PLC-Variablen-tabelle muss dem Namen der Sondervariable exakt entsprechen, wenn keine Use-Klausel angegeben ist.

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use=' "Status" ' -->
```

In diesem Beispiel empfängt die HTTP-Sondervariable "HEADER:Status" den Wert der PLC-Variable "Status".

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablennamen mit Sonderzeichen (Seite 565) beschrieben.

### 11.3.2.4 Sondervariablen schreiben

Der Webserver bietet die Möglichkeit, Werte aus Sondervariablen der HTTP-Antwortkopfzeile in die CPU zu schreiben. Sie können beispielsweise in STEP 7 Informationen zu dem Cookie einer benutzerdefinierten Webseite, zu dem Benutzer, der auf eine Seite zugreift, oder Header-Informationen speichern. Der Webserver bietet Zugriff auf bestimmte Sondervariablen, die Sie, wenn Sie als admin-Benutzer angemeldet sind, in die CPU schreiben können.

#### Syntax

```
<!-- AWP_In_Variable Name='<Typ>:<Name>' [Use='<Varname>'] -->
```

#### Parameter

<Typ>	Der Typ der Sondervariable. Möglich sind: HEADER SERVER COOKIE_VALUE
<Name>	Sondervariable innerhalb der oben definierten Typen, wie in diesen Beispielen gezeigt: HEADER:Accept: akzeptable Inhaltstypene HEADER:User-Agent: Informationen zum Benutzeragenten, von dem die Anforderung stammt. SERVER:current_user_id: ID des aktuellen Benutzers; 0, wenn kein Benutzer angemeldet ist SERVER:current_user_name: Name des aktuellen Benutzers COOKIE_VALUE:<name>: Wert des genannten Cookies Setzen Sie die Namensklausele in einfache Anführungszeichen. Ist keine Verwendungsklausele angegeben, entspricht der Sondervariablenname einem PLC-Variablennamen. Setzen Sie die vollständige Namensklausele in einfache Anführungszeichen und die PLC-Variable in doppelte Anführungszeichen. Der Sondervariablenname muss dem PLC-Variablennamen exakt entsprechen. In der HTTP-Dokumentation finden Sie eine Liste aller Namen der HEADER-Variablen.
<Varname>	Der Variablenname in Ihrem STEP 7-Programm, in das Sie die Sondervariable schreiben möchten. Es kann sich um eine PLC-Variable oder eine Datenbausteinvariable handeln. Der Variablenname muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinamen in doppelte Anführungszeichen. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden.

#### Beispiele

```
<!-- AWP_In_Variable Name='\"SERVER:current_user_id\"' -->
```

In diesem Beispiel schreibt die Webseite den Wert der HTTP-Sondervariablen "SERVER:current\_user\_id" in die PLC-Variable "SERVER:current\_user\_id".

```
<!-- AWP_In_Variable Name=SERVER:current_user_id'
Use=' "Meine_Benutzer-ID" ' -->
```

In diesem Beispiel schreibt die Webseite den Wert der HTTP-Sondervariablen "SERVER:current\_user\_id" in die PLC-Variable "Meine\_Benutzer-ID".

---

**Hinweis**

Nur der admin-Benutzer kann Daten in die CPU schreiben. Die Befehle werden ignoriert, wenn sich der Benutzer nicht als admin angemeldet hat.

---

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter "Handhabung von Variablennamen mit Sonderzeichen (Seite 565)" beschrieben.

**11.3.2.5 Alias für einen Variablenverweis nutzen**

Für eine In\_Variable oder eine Out\_Variable können Sie in Ihren benutzerdefinierten Webseiten einen Alias verwenden. Sie können beispielsweise einen anderen symbolischen Namen in Ihrer HTML-Seite verwenden als den in der CPU, oder Sie können eine Variable in der CPU einer Sondervariablen gleichsetzen. Die AWP-Verwendungsklausel bietet diese Möglichkeit.

**Syntax**

```
<-- AWP_In_Variable Name='<Varname1>' Use='<Varname2>' -->
<-- AWP_Out_Variable Name='<Varname1>' Use='<Varname2>' -->
```

**Parameter**

<Varname1>	Der Aliasname oder der Sondervariablenname Varname1 muss in einfachen oder doppelten Anführungszeichen angegeben werden.
<Varname2>	Name der PLC-Variable, der Sie einen Aliasnamen zuweisen möchten. Bei der Variable kann es sich um eine PLC-Variable, eine Datenbausteinvariable oder eine Sondervariable handeln. Varname2 muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable, Sondervariable oder einen Datenbausteinnamen in doppelte Anführungszeichen. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden.

**Beispiele**

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use=' "Data_Block_10".server_user' -->
```



In diesem Beispiel wird die Sondervariable SERVER:current\_user\_id in die Variable "server\_user" im Datenbaustein "Data\_Block\_10" geschrieben.

```
<-- AWP_Out_Variable Name='Weight'
Use=' "Data_Block_10".Tank_data.Weight' -->
```

In diesem Beispiel kann der Wert in Datenbaustein-Strukturelement Data\_Block\_10.Tank\_data.Weight innerhalb der übrigen benutzerdefinierten Webseite einfach als "Weight" referenziert werden.

```
<-- AWP_Out_Variable Name='Gewicht' Use=' "Gewicht_Rohmilchbehälter" '
-->
```

In diesem Beispiel kann der Wert der PLC-Variable "Gewicht\_Rohmilchbehälter" in der übrigen benutzerdefinierten Webseite einfach als "Gewicht" angegeben werden.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablenamen mit Sonderzeichen (Seite 565) beschrieben.

### 11.3.2.6 Enum-Typen definieren

Sie können in Ihren benutzerdefinierten Seiten Enum-Typen definieren und die Elemente in einem AWP-Befehl zuweisen.

#### Syntax

```
<!-- AWP_Enum_Def Name='<Name Enum-Typ>' Values='<Wert>, <Wert>, ...
' -->
```

#### Parameter

<Name Enum-Typ>	Name des Aufzählungstyps, in einfache oder doppelte Anführungszeichen gesetzt.
<Wert>	<Konstante>:<Name> Die Konstante kennzeichnet den numerischen Wert für die Zuweisung des Enum-Typs. Die Gesamtzahl ist unbegrenzt. Der Name ist der dem Enum-Element zugewiesene Wert.

Beachten Sie, dass die gesamte Zeichenkette der Enum-Wertzuweisungen in einzelne Hochkommata eingeschlossen ist, und jede einzelne Elementzuweisung zu einem Enum-Typ steht in doppelten Anführungszeichen. Die Definition eines Enum-Typs hat für die benutzerdefinierten Webseiten globale Gültigkeit. Wenn Sie Ihre benutzerdefinierten Webseiten in sprachspezifischen Ordnern (Seite 584) abgelegt haben, gilt die Enum-Typ-Definition global für alle Seiten in dem Sprachordner.

#### Beispiel

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Keine Alarme",
1:"Behälter ist voll", 2:"Tank is empty"' -->
```

### 11.3.2.7 CPU-Variablen mit einem Enum-Typ referenzieren

Sie können eine Variable in der CPU einem Enum-Typ zuweisen. Diese Variable kann an anderer Stelle in Ihren benutzerdefinierten Webseiten in einer Leseoperation (Seite 554) oder einer Schreiboperation (Seite 555) verwendet werden. Bei einer Leseoperation ersetzt der Webserver den aus der CPU gelesenen numerischen Wert durch den entsprechenden Enum-Textwert. Bei einer Schreiboperation ersetzt der Webserver den Textwert mit dem ganzzahligen Wert der Aufzählung, der dem Text entspricht, bevor der Wert in die CPU geschrieben wird.

#### Syntax

```
<!-- AWP_Enum_Ref Name='<Varname>' Enum="<Enum-Typ>" -->
```

#### Parameter

<Varname>	Name der PLC-Variable oder Datenbausteinvariable, die dem Enum-Typ zugewiesen werden soll, oder Name des Aliasnamens einer PLC-Variable (Seite 560), sofern deklariert.  Varname muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinnamen in doppelte Anführungszeichen. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins.
<Enum-Typ>	Name des Aufzählungstyps, in einfache oder doppelte Anführungszeichen gesetzt

Die Referenz eines Enum-Typs gilt für das aktuelle Fragment.

#### Beispiel für eine Deklaration

```
<!-- AWP_Enum_Ref Name="Alarm" Enum="AlarmEnum" -->
```

#### Beispiel für die Verwendung beim Lesen einer Variablen

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Keine Alarme",
1:"Behälter ist voll", 2:"Behälter ist leer"' -->
<!-- AWP_Enum_Ref Name="Alarm" Enum="AlarmEnum" -->
...
<p>Der aktuelle Wert von "Alarm" ist := "Alarm":</p>
```

Wenn der Wert von "Alarm" in der CPU gleich 2 ist, zeigt die HTML-Seite 'Der aktuelle Wert von "Alarm" ist Behälter ist leer' an, weil die Definition des Enum-Typs (Seite 561) dem numerischen Wert 2 die Zeichenfolge "Behälter ist leer" zuweist.

#### Beispiel für die Verwendung beim Schreiben einer Variablen

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Keine Alarme",
1:"Behälter ist voll", 2:"Behälter ist leer"' -->
<!-- AWP_In_Variable Name="Alarm" -->
<!-- AWP_Enum_Ref Name="Alarm" Enum="AlarmEnum" -->
...
<form method="POST">
```

```
<p><input type="hidden" name=' "Alarm"' value="Behälter ist voll"
/></p>
<p><input type="submit" value='Behälter ist voll setzen' /><p>
</form>
```

Weil die Definition des Enum-Typs (Seite 561) dem numerischen Wert 1 den Text "Behälter ist voll" zuweist, wird der Wert 1 in die PLC-Variablen "Alarm" in der CPU geschrieben.

Beachten Sie, dass die Namensklausele in der Deklaration AWP\_In\_Variable exakt der Namensklausele in der Deklaration AWP\_Enum\_Ref entsprechen muss.

### Beispiel für das Schreiben von Variablen unter Verwendung von Aliasnamen

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Keine Alarme",
1:"Behälter ist voll", 2:"Behälter ist leer"' -->
<!-- AWP_In_Variable Name=' "Alarm"'
Use=' "Data_block_4".Motor1.Alarm'-->
<!-- AWP_Enum_Ref Name=' "Alarm"' Enum='AlarmEnum' -->
...
<form method="POST">
<p><input type="hidden" name=' "Alarm"' value="Behälter ist voll"
/></p>
<p><input type="submit" value='Behälter ist voll setzen' /><p>
</form>
```

Weil die Definition des Enum-Typs (Seite 561) dem numerischen Wert 1 den Text "Behälter ist voll" zuweist, wird der Wert 1 in den Aliasnamen "Alarm" geschrieben, welcher der PLC-Variablen "Motor1.Alarm" im Datenbaustein "Data\_Block\_4" der CPU entspricht.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablennamen mit Sonderzeichen (Seite 565) beschrieben.

### 11.3.2.8 Fragmente erstellen

STEP 7 konvertiert und speichert benutzerdefinierte Webseiten als Steuerungs-DB und DB-Fragmente, wenn Sie in den CPU-Eigenschaften für den Webserver auf "Bausteine generieren" klicken. Sie können spezifische Fragmente für spezifische Seiten oder für Abschnitte spezifischer Seiten einrichten. Sie können diese Fragmente mit dem AWP-Befehl "Start\_Fragment" durch einen Namen und eine Nummer kennzeichnen. Alles auf einer Seite, was auf den Befehl AWP\_Start\_Fragment folgt, gehört zu dem Fragment, bis ein nachfolgender Befehl AWP\_Start\_Command abgesetzt wird oder bis das Ende der Datei erreicht ist.

#### Syntax

```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Typ>] [ID=<ID>] -->
```

**Parameter**

<Name>	Text-Zeichenkette: Name des DB-Fragments Fragmentnamen müssen mit einem Buchstaben oder einem Unterstrich beginnen und sich aus Buchstaben, Ziffern und Unterstrichen zusammensetzen. Der Fragmentname ist ein regulärer Ausdruck in der Form: [a-zA-Z_][a-zA-Z_0-9]*
<Typ>	"Manuell" oder "Automatisch" Manuell: Das STEP 7-Programm muss dieses Fragment anfordern und kann entsprechend reagieren. Die Funktionsweise des Fragments muss mit STEP 7 und den Variablen des Steuerungs-DBs gesteuert werden. Automatisch: Der Webserver verarbeitet das Fragment automatisch. Wenn Sie den Parameter für den Typ nicht angeben, ist die Voreinstellung "Automatisch".
<ID>	Ganzzahlige Identifikationsnummer. Wenn Sie den Parameter ID nicht angeben, weist der Webserver standardmäßig eine Nummer zu. Geben Sie bei manuellen Fragmenten für die ID eine niedrige Nummer an. Die ID ist der Weg, über den das STEP 7-Programm ein manuelles Fragment steuert.
<Modus>	"sichtbar" oder "ausgeblendet" sichtbar: Inhalte des Fragments werden auf der benutzerdefinierten Webseite angezeigt. ausgeblendet: Inhalte des Fragments werden nicht auf der benutzerdefinierten Webseite angezeigt. Wenn Sie den Parameter für den Typ nicht angeben, ist die Voreinstellung "sichtbar".

**Manuelle Fragmente**

Wenn Sie ein manuelles Fragment für eine benutzerdefinierte Webseite oder einen Teil einer Seite anlegen, muss Ihr STEP 7-Programm steuern, wann das Fragment gesendet wird. Das STEP 7-Programm muss im Steuer-DB für eine benutzerdefinierte Seite unter manueller Steuerung entsprechende Parameter festlegen und dann die WWW-Anweisung mit dem geänderten Steuer-DB aufrufen. Nähere Erläuterungen zur Struktur des Steuer-DBs und dazu, wie Sie einzelne Seiten und Fragmente ändern, finden Sie unter "Erweiterte Steuerung von benutzerdefinierten Webseiten (Seite 587)".

**11.3.2.9 Fragmente importieren**

Sie können von einem Teil Ihres HTML-Codes ein benanntes Fragment erstellen und dann dieses Fragment an anderer Stelle in Ihren Satz benutzerdefinierter Webseiten importieren. Stellen Sie sich z. B. einen Satz benutzerdefinierter Webseiten vor, die aus einer Startseite und verschiedenen anderen HTML-Seiten bestehen, die über Links auf der Startseite aufgerufen werden. Angenommen, auf jeder dieser einzelnen Seiten soll das Firmenlogo angezeigt werden. Dafür könnten Sie ein Fragment erstellen (Seite 563), das das Bild des Firmenlogos lädt. Jede einzelne HTML-Seite kann dann dieses Fragment importieren, um das Firmenlogo anzuzeigen. Für diesen Zweck verwenden Sie den AWP-Befehl `Import_Fragment`. Der HTML-Code für das Fragment ist nur in einem Fragment vorhanden, doch Sie können dieses DB-Fragment so oft wie nötig in so viele Webseiten, wie Sie wünschen, importieren.

## Syntax

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

## Parameter

<Name>	Text-Zeichenkette: Name des zu importierenden DB-Fragments
--------	--

## Beispiel

Auszug aus dem HTML-Code, der ein Fragment zum Anzeigen eines Bilds erstellt:

```
<!-- AWP_Start_Fragment Name='Mein_Firmenlogo' --><p></p>
```

Auszug aus dem HTML-Code in einer anderen \*.html-Datei, der das Fragment mit dem Logobild importiert:

```
<!-- AWP_Import_Fragment Name='Mein_Firmenlogo' -->
```

Beide \*.html-Dateien (die Datei, die das Fragment erstellt und die Datei, die das Fragment importiert) befinden sich in der Ordnerstruktur, die Sie festlegen, wenn Sie die benutzerdefinierten Seiten in STEP 7 konfigurieren (Seite 567).

### 11.3.2.10 Definitionen verbinden

Wenn Sie die Variablen für die Verwendung in Ihren benutzerdefinierten Webseiten deklarieren, können Sie eine Variablendeklaration mit einem Alias für die Variable (Seite 560) verbinden. Sie können außerdem mehrere In\_Variables in einer Anweisung und mehrere Out\_Variables in einer Anweisung verbinden.

## Beispiele

```
<!-- AWP_In_Variable Name='"Füllstand"', Name='"Gewicht"',
Name='"Temp"' -->
<--! AWP_Out_Variable Name='HEADER:Status', Use='"Status"',
Name='HEADER:Location', Use="Speicherort",
Name='COOKIE_VALUE:name', Use="Mein_Cookie" -->
<!-- AWP_In_Variable Name='Alarm' Use='"Datenbaustein_10".Alarm' -->
```

### 11.3.2.11 Handhabung von Variablennamen mit Sonderzeichen

Bei der Angabe von Variablennamen in benutzerdefinierten Webseiten müssen Sie mit besonderer Vorsicht vorgehen, wenn die Variablennamen Zeichen mit besonderer Bedeutung enthalten.

## Variablen lesen

Um eine Variable zu lesen (Seite 554), verwenden Sie die folgende Syntax:

```
:=<Varname>:
```

Die folgenden Regeln gelten beim Lesen von Variablen:

- Bei Variablenamen aus der PLC-Variablen-tabelle setzen Sie den Variablenamen in doppelte Anführungszeichen.
- Bei Variablenamen, bei denen es sich um Datenbausteinvariablen handelt, ist der Datenbausteinname in doppelte Anführungszeichen zu setzen. Die Variable befindet sich außerhalb der Anführungszeichen.
- Bei Variablenamen, bei denen es sich um direkte E/A-Adressen, Speicheradressen oder Aliasnamen handelt, setzen Sie die gelesene Variable nicht in Anführungszeichen.
- Bei Variablenamen oder Datenbausteinvariablenamen, die einen nach links geneigten Schrägstrich ("Backslash") enthalten, stellen Sie dem Backslash einen weiteren Backslash voran.
- Wenn ein Variablenname oder Datenbausteinvariablenname einen Doppelpunkt, ein Kleiner-als-Zeichen, Größer-als-Zeichen oder ein Kaufmanns-Und enthält, definieren Sie für die gelesene Variable einen Alias ohne Sonderzeichen und lesen die Variable über diesen Aliasnamen. Stellen Sie in Variablenamen enthaltenen Doppelpunkten in Verwendungsklauseln einen Backslash voran.

Tabelle 11- 1 Beispiele für zu lesende Variablen

Datenbausteinname	Variablenname	Lesebefehl
-/-	ABC:DEF	<code>&lt;!--AWP_Out_Variable Name='Sondervariable' Use ='"ABC:DEF"' --&gt; :=Sondervariable:</code>
nicht zutreffend	T\	<code>:= "T\\" :</code>
nicht zutreffend	A \B 'C :D	<code>&lt;!--AWP_Out_Variable Name='Weitere_Sondervariable' Use='\"A \\B \\ 'C :D\"' --&gt; :=Weitere_Sondervariable:</code>
nicht zutreffend	a<b	<code>&lt;!--AWP_Out_Variable Name='a_kleiner_b' Use='\"a&lt;b\"' --&gt; :=a_kleiner_b:</code>
Datenbaustein_1	Variable_1	<code>:= "Datenbaustein_1".Tag_1:</code>
Datenbaustein_1	ABC:DEF	<code>&lt;!-- AWP_Out_Variable Name='Sondervariable' Use='\"Datenbaustein_1\".ABC\ :DEF' --&gt; :=Sondervariable:</code>
DB A' B C D\$ E	Variable	<code>:= "DB A' B C D\$ E".Tag:</code>
DB:DB	Variable:Variable	<code>&lt;!--AWP_Out_Variable Name='meine_Variable' Use = '\"DB:DB\".Tag\ :Tag' --&gt; :=meine_Variable:</code>

## Namens- und Verwendungsklauseln

Die AWP-Befehle AWP\_In\_Variable, AWP\_Out\_Variable, AWP\_Enum\_Def, AWP\_Enum\_Ref, AWP\_Start\_Fragment und AWP\_Import\_Fragment haben Namensklauseln. HTML-Formularbefehle wie <input> und <select> haben ebenfalls Namensklauseln. AWP\_In\_Variable und AWP\_Out\_Variable können zusätzlich Verwendungsklauseln haben. Unabhängig vom Befehl ist die Syntax von Namens- und Verwendungsklauseln hinsichtlich der Handhabung von Sonderzeichen die gleiche:

- Der Text, den Sie für eine Namens- oder Verwendungsklausel angeben, ist in einfache Anführungszeichen zu setzen. Handelt es sich bei dem Namen in Anführungszeichen um eine PLC-Variable oder einen Datenbausteinname, setzen Sie die gesamte Klausel in einfache Anführungszeichen.
- Innerhalb einer Namens- oder Verwendungsklausel sind Datenbausteinname und PLC-Variablenname in doppelte Anführungszeichen zu setzen.
- Wenn ein Variablenname oder Datenbausteinname ein einfaches Anführungszeichen oder einen rückwärts gerichteten Schrägstrich ("Backslash") enthält, stellen Sie dem Zeichen einen Backslash als Escape-Zeichen voran. Der rückwärts gerichtete Schrägstrich "\" dient bei der Übersetzung der AWP-Befehle als Escape-Zeichen.

Tabelle 11- 2 Beispiele für Namensklauseln

Datenbausteinname	Variablenname	Möglichkeiten für Namensklauseln
-/-	ABC'DEF	Name=" 'ABC\ 'DEF" '
nicht zutreffend	A \B 'C :D	Name=" 'A \\B \ 'C :D" '
Datenbaustein_1	Variable_1	Name=" "Datenbaustein_1".Tag_1'
Datenbaustein_1	ABC'DEF	Name=" "Datenbaustein_1".ABC\ 'DEF'
Datenbaustein_1	A \B 'C :D	Name=" "Datenbaustein_1".A \\B \ 'C :D'
DB A' B C D\$ E	Variable	Name=" "DB A\ ' B C D\$ E".Variable'

Für Verwendungsklauseln gelten die gleichen Konventionen wie für Namensklauseln.

### Hinweis

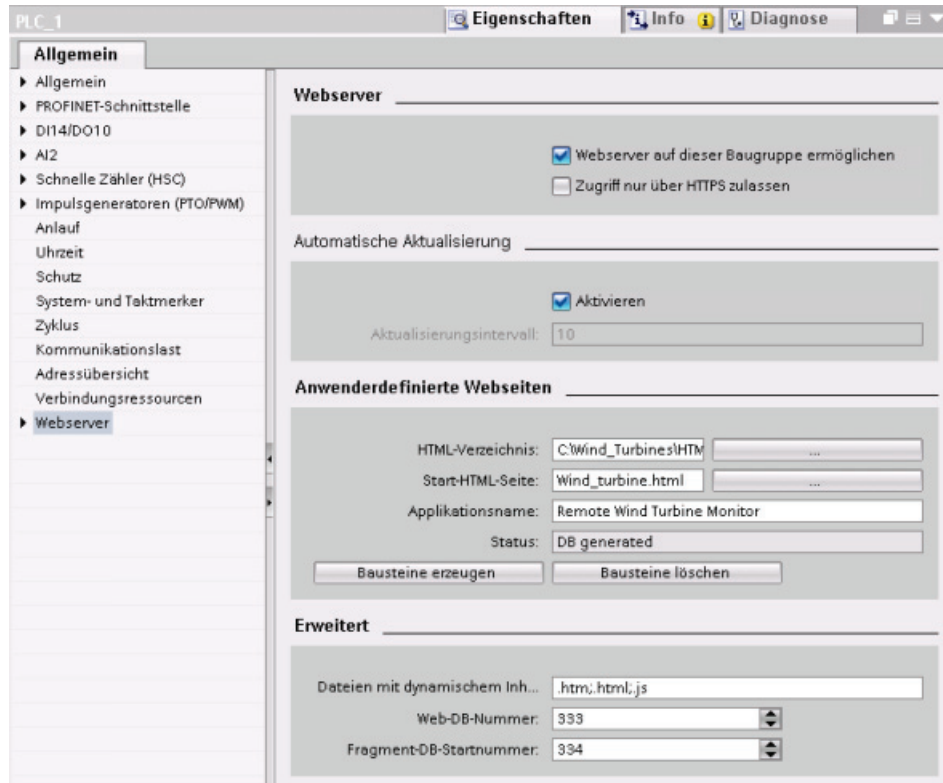
Unabhängig von den Zeichen, die Sie in Ihrer HTML-Seite verwenden, legen Sie als Zeichensatz der HTML-Seite UTF-8 fest und speichern die Seite im Editor mit der Zeichenverschlüsselung UTF-8.

### 11.3.3 Verwendung von benutzerdefinierten Webseiten konfigurieren

Um benutzerdefinierte Webseiten in STEP 7 zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die CPU in der Gerätekonfiguration aus.

2. Rufen Sie im Inspektorfenster der CPU die Eigenschaften des Webserver auf.



3. Sofern nicht bereits geschehen, aktivieren Sie das Kontrollkästchen für "Webserver aktivieren".
4. Aktivieren Sie das Kontrollkästchen "Erlaubt nur HTTPS-Zugriff", um sicherzustellen, dass der Webserver die Kommunikation verschlüsselt und um die Sicherheit Ihrer über das Web zugänglichen CPU zu erhöhen.
5. Geben Sie den Namen des Ordners auf Ihrem PC ein, in dem Sie die HTML-Standardseite (Startseite) gespeichert haben.
6. Geben Sie den Namen der Standardseite ein.
7. Geben Sie einen Namen für Ihre Anwendung ein (optional). Der Anwendungsname dient zur weiteren Unterteilung bzw. Gruppierung der Webseiten. Wenn ein Anwendungsname vorhanden ist, wird die URL in folgendem Format angezeigt:  
http://ww.xx.yy.zz/awp/<Anwendungsname>/<Seitenname>.html.
8. Geben Sie an, welche Dateierweiterungen auf das Vorhandensein von AWP-Befehlen zu prüfen sind. Standardmäßig analysiert STEP 7 Dateien mit den Erweiterungen \*.htm, \*.html und \*.js. Wenn Sie andere Dateierweiterungen nutzen, fügen Sie sie ein.
9. Übernehmen Sie die Standardnummer für den Web-DB oder geben Sie eine Nummer Ihrer Wahl ein. Dies ist die DB-Nummer des Steuer-DBs, der die Anzeige der Webseiten steuert.
10. Übernehmen Sie die standardmäßige Anfangsnummer für das DB-Fragment oder geben Sie eine Nummer Ihrer Wahl ein. Dies ist das erste der DB-Fragmente mit den Webseiten.



### Programmbausteine generieren

Wenn Sie auf die Schaltfläche "Bausteine generieren" klicken, generiert STEP 7 aus den HTML-Seiten im von Ihnen angegebenen HTML-Quellverzeichnis Datenbausteine und einen Steuerdatenbaustein für den Betrieb Ihrer Webseiten. Sie können diese Attribute nach Bedarf für Ihre Anwendung festlegen (Seite 569). STEP 7 generiert außerdem einen Satz Datenbausteinfragmente, um die Darstellung aller Ihrer HTML-Seiten zu speichern. Wenn Sie die Datenbausteine generieren, aktualisiert STEP 7 die Eigenschaften, um die Nummer des Steuerdatenbausteins und die Nummer des ersten Datenbausteinfragments anzuzeigen. Nachdem Sie die Datenbausteine generiert haben, sind Ihre benutzerdefinierten Webseiten Teil Ihres STEP 7-Programms. Die diesen Seiten entsprechenden Bausteine erscheinen im Webserverordner, der sich in der Projektnavigation unter den Programmbausteinen im Ordner "Systembausteine" befindet.

### Programmbausteine löschen

Um Datenbausteine zu löschen, die Sie zuvor generiert haben, klicken Sie auf die Schaltfläche "Datenbausteine löschen". STEP 7 löscht den Steuerdatenbaustein und alle Datenbausteinfragmente aus Ihrem Projekt, in dem sich die benutzerdefinierten Webseiten befinden.


## 11.3.4 WWW-Anweisung für benutzerdefinierte Webseiten programmieren

Die WWW-Anweisung muss in Ihrem STEP 7-Anwenderprogramm enthalten sein und ausgeführt werden, damit die benutzerdefinierten Webseiten über die Standard-Webseiten aufrufbar sind. Der Steuerdatenbaustein ist der Eingangsparameter für die WWW-Anweisung und gibt den Inhalt der Seiten wie in den Datenbausteinfragmenten dargestellt sowie Zustands- und Steuerinformationen an. STEP 7 erstellt den Steuerdatenbaustein, wenn Sie in der Konfiguration der benutzerdefinierten Webseiten (Seite 567) auf die Schaltfläche "Bausteine erstellen" klicken.

### WWW-Anweisung programmieren

Das STEP 7-Programm muss die Anweisung WWW ausführen, damit die benutzerdefinierten Webseiten über die Standard-Webseiten aufrufbar sind. Sie können festlegen, dass die benutzerdefinierten Webseiten nur unter bestimmten Bedingungen entsprechend den Anwendungsvoraussetzungen und Einstellungen verfügbar sind. Dann kann Ihre Programmlogik steuern, wann die Anweisung WWW aufzurufen ist.

Tabelle 11- 3 Anweisung WWW

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := WWW(     ctrl_db:=_uint_in_);</pre>	Zugriff auf die benutzerdefinierten Webseiten über die Standard-Webseiten

11.3 Benutzerdefinierte Webseiten

Sie müssen den Eingangsparameter des Steuerdatenbausteins (CTRL\_DB), der der ganzzahligen DB-Nummer des Steuer-DBs entspricht, eingeben. Sie finden diese Bausteinnummer des Steuer-DBs (als Web-DB-Nummer bezeichnet) in den Webserver-Eigenschaften der CPU, nachdem Sie die Bausteine für die benutzerdefinierten Webseiten erstellt haben. Geben Sie die ganzzahlige DB-Nummer als Parameter CTRL\_DB der Anweisung WWW an. Der Rückgabewert (RET\_VAL) enthält das Funktionsergebnis. Beachten Sie, dass die Anweisung WWW asynchron ausgeführt wird und dass der Ausgang RET\_VAL einen Anfangswert von 0 hat, auch wenn später ein Fehler auftreten kann. Das Programm kann den Zustand des Steuer-DBs abfragen, um sicherzustellen, dass die Anwendung erfolgreich gestartet wurde, oder es kann mit einem nachfolgenden Aufruf von WWW den Parameter RET\_VAL abfragen.

Tabelle 11- 4 Rückgabewert

RET_VAL	Beschreibung
0	Kein Fehler
16#00yx	x: Die vom entsprechenden Bit dargestellte Anforderung ist im Wartezustand: x=1: Anforderung 0 x=2: Anforderung 1 x=4: Anforderung 2 x=8: Anforderung 3 Die x-Werte können logisch durch ODER verknüpft werden, um die Wartezustände mehrerer Anforderungen darzustellen. Wenn z. B. x = 6 ist, sind die Anforderungen 1 und 2 im Wartezustand. y: 0: kein Fehler; 1: Fehler vorhanden und "last_error" wurde im Steuer-DB gesetzt
16#803a	Der Steuer-DB ist nicht geladen.
16#8081	Datentyp, Format oder Version des Steuer-DBs ist falsch.
16#80C1	Für die Initialisierung der Webanwendung sind keine Ressourcen verfügbar.

**Verwendung des Steuer-DBs**

STEP 7 erstellt den Steuerdatenbaustein, wenn Sie auf die Schaltfläche "Bausteine generieren" klicken. Die Nummer des Steuer-DBs wird in den Eigenschaften der benutzerdefinierten Webseiten angezeigt. Sie finden den Steuer-DB auch im Ordner "Programmbausteine" in der Projektnavigation.

Typischerweise verwendet Ihr STEP 7-Programm den Steuer-DB direkt wie über den Vorgang "Bausteine generieren" angelegt, ohne weitere Änderungen. Das STEP 7-Anwenderprogramm kann jedoch globale Befehle im Steuer-DB festlegen, um den Webserver zu deaktivieren oder ihn nachfolgend wieder zu aktivieren. Außerdem muss das STEP 7-Anwenderprogramm das Verhalten von benutzerdefinierten Webseiten, die Sie als manuelle DB-Fragmente anlegen (Seite 567), über eine Anforderungstabelle im Steuer-DB steuern. Informationen zu diesen erweiterten Aufgaben finden Sie unter Erweiterte Steuerung von benutzerdefinierten Webseiten (Seite 587).

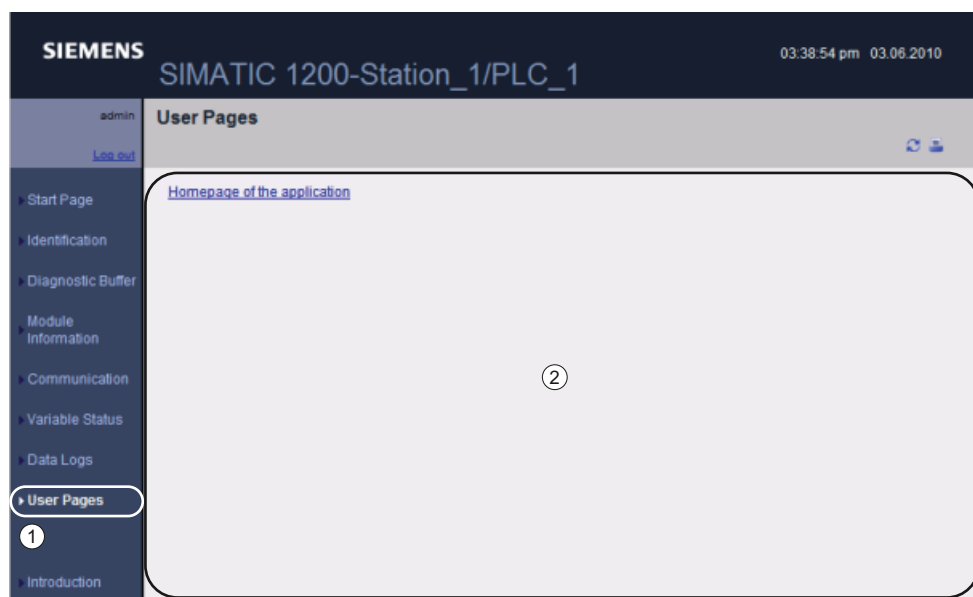
### 11.3.5 Programmbausteine in die CPU laden

Nachdem Sie die Bausteine für die benutzerdefinierten Webseiten generiert haben, sind diese Teil Ihres STEP 7-Programms, genau wie alle anderen Programmbausteine. Um die Programmbausteine in die CPU zu laden, gehen Sie wie üblich vor. Beachten Sie, dass Sie Programmbausteine für benutzerdefinierte Webseiten nur in die CPU laden können, wenn sich diese im Betriebszustand STOP befindet.

### 11.3.6 Über den PC auf die benutzerdefinierten Webseiten zugreifen

Sie rufen die benutzerdefinierten Webseiten über die Standard-Webseiten (Seite 533) auf. Die Standard-Webseiten zeigen einen Link für die Benutzerseiten auf der linken Seite an, wo sich auch die Links auf die anderen Seiten befinden. Wenn Sie auf den Link "Benutzerseiten" klicken, ruft Ihr Webbrowser die Seite auf, die Zugriff auf Ihre Standardseite bietet. Innerhalb der benutzerdefinierten Seiten entspricht die Navigation dem Aufbau Ihrer spezifischen Seiten.

Der Webserver zeigt den Inhalt der benutzerdefinierten Seiten innerhalb des Rahmens der Standard-Webseiten an. Das heißt, der Kopfbereich oben sowie die Anmelde- und Navigationsbereiche links bleiben an ihren vorgegebenen Positionen.



- ① Link zur Standardseite der benutzerdefinierten Webseiten
- ② Inhaltsbereich der benutzerdefinierten Webseiten

### 11.3.7 Einschränkungen bei benutzerdefinierten Webseiten

Die Einschränkungen bei Standard-Webseiten (Seite 591) gelten auch bei benutzerdefinierten Webseiten. Zudem gibt es bei benutzerdefinierten Webseiten einige spezifische Aspekte.

### Kapazität des Ladespeichers

Ihre benutzerdefinierten Webseiten werden, wenn Sie auf "Bausteine generieren" klicken, zu Datenbausteinen, die Platz im Ladespeicher benötigen. Wenn Sie eine Memory Card gesteckt haben, steht die Kapazität Ihrer Memory Card als externer Ladespeicher für die benutzerdefinierten Webseiten zur Verfügung.

Wenn Sie keine Memory Card gesteckt haben, belegen diese Bausteine Platz im internen Ladespeicher, der je nach CPU-Modell begrenzt ist.

Mit den Online- und Diagnosefunktionen in STEP 7 können Sie den belegten und den freien Platz im Ladespeicher abfragen. Sie können zudem in den Eigenschaften der einzelnen Bausteine, die STEP 7 für Ihre benutzerdefinierten Webseiten generiert, den benötigten Platz im Ladespeicher prüfen.

---

#### Hinweis

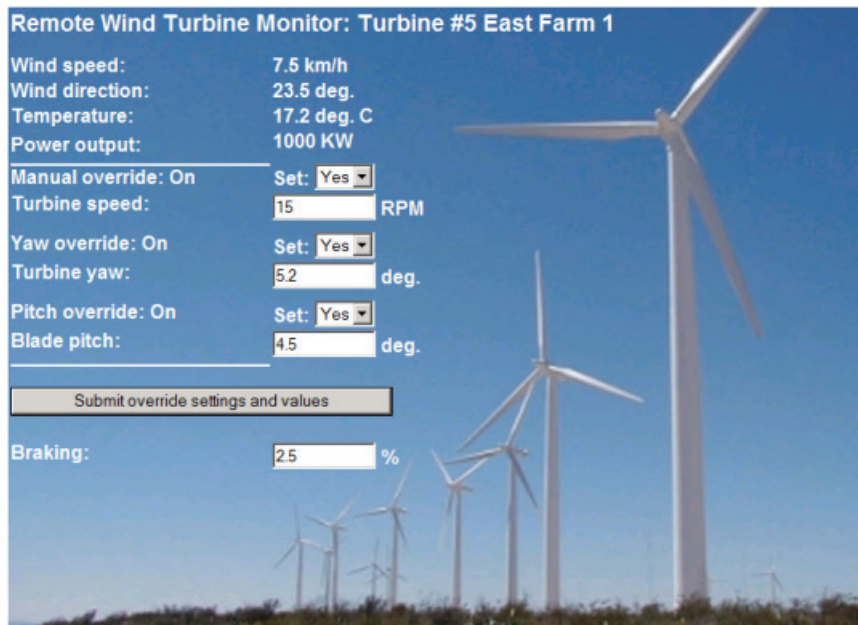
Wenn Sie den Platz für Ihre benutzerdefinierten Webseiten verringern müssen, entfernen Sie ggf. einige der eingefügten Bilder.

---

## 11.3.8 Beispiel für eine benutzerdefinierte Webseite

### 11.3.8.1 Webseite zum Beobachten und Steuern einer Windturbine

Stellen Sie sich als Beispiel für eine benutzerdefinierte Webseite eine Webseite vor, die dazu dient, eine Windturbine entfernt zu beobachten und zu steuern:



### Hinweis

Diese Beispielseite ist in Englisch, doch Sie können natürlich bei der Entwicklung Ihrer eigenen HTML-Seiten jede beliebige Sprache verwenden.

---

## Beschreibung

In dieser Anwendung ist jede Windturbine des Windparks mit einer S7-1200 zur Steuerung der Turbine ausgestattet. Im STEP 7-Programm hat jede Windturbine einen Datenbaustein mit Daten, die für die jeweilige Windturbine und den jeweiligen Standort spezifisch sind.

Die benutzerdefinierte Webseite bietet dezentralen Zugriff auf die Turbine über einen PC. Ein Benutzer kann die Standard-Webseiten der CPU einer bestimmten Windturbine aufrufen und auf die benutzerdefinierte "Remote Wind Turbine Monitor"-Webseite (Webseite für die dezentrale Beobachtung der Windturbine) zugreifen, um die Daten der Turbine einzusehen. Ein admin-Benutzer kann zudem die Turbine in den manuellen Modus versetzen und die Variablen für Drehzahl, Ausrichtung und Anstellwinkel der Turbine über die Webseite steuern. Ein admin-Benutzer kann auch einen Bremswert festlegen, unabhängig davon, ob die Turbine manuell oder automatisch gesteuert wird.

Das STEP 7-Programm prüft die Booleschen Werte für Übersteuerung der automatischen Steuerung und verwendet, sofern diese eingestellt ist, die vom Anwender eingegebenen Werte für Drehzahl, Ausrichtung und Anstellwinkel der Turbine. Andernfalls ignoriert das Programm diese Werte.

## Verwendete Dateien

Dieses Beispiel für eine benutzerdefinierte Webseite besteht aus drei Dateien:

- **Wind\_turbine.html:** Dies ist die HTML-Seite mit der oben dargestellten Anzeige. Über AWP-Befehle wird auf Steuerungsdaten zugegriffen.
- **Wind\_turbine.css:** Dies ist das Cascading Style Sheet, das die Formatierungsvorgaben für die HTML-Seite enthält. Die Verwendung eines Cascading Style Sheet ist optional, doch es kann die Entwicklung von HTML-Seiten vereinfachen.
- **Wind\_turbine.jpg:** Dies ist das Hintergrundbild, das auf der HTML-Seite angezeigt wird. Die Verwendung von Bildern in benutzerdefinierten Webseiten ist natürlich freigestellt, und Bilder benötigen zusätzlichen Speicherplatz in der CPU.

Diese Dateien sind nicht in Ihrer Installation enthalten, werden jedoch als Beispiel beschrieben.

## Implementierung

Die HTML-Seite verwendet AWP-Befehle zum Auslesen von Werten aus dem PLC-Gerät (Seite 554) für die Anzeigefelder und sie verwendet AWP-Befehle zum Schreiben von Werten in das PLC-Gerät (Seite 555) für die Daten der Benutzereingabe. Diese Seite nutzt zudem AWP-Befehle für die Definition von Enum-Typen (Seite 561) und für die Referenz (Seite 562) zur Handhabung von EIN/AUS-Einstellungen.

Der erste Teil der Seite zeigt eine Kopfzeile mit der Nummer und dem Standort der Windturbine an.

**Remote Wind Turbine Monitor: Turbine #5 East Farm 1**

Der nächste Teil der Seite zeigt die atmosphärischen Bedingungen an der Windturbine an. Diese Felder werden von den E/A am Turbinenstandort geliefert, die die Windgeschwindigkeit, Windrichtung und aktuelle Temperatur bereitstellen.

Wind speed:	7.5 km/h
Wind direction:	23.5 deg.
Temperature:	17.2 deg. C

Dann zeigt die Seite die aus der S7-1200 ausgelesene Leistungsabgabe der Turbine an.

Power output:	1000 KW
---------------	---------

Die folgenden Abschnitte ermöglichen die manuelle Steuerung der Turbine, eine Übersteuerung der normalen Automatiksteuerung der S7-1200. Die folgenden Übersteuerungen sind möglich:

- Manuelle Übersteuerung: Aktiviert die manuelle Übersteuerung der Turbine. Das STEP 7-Anwenderprogramm verlangt, dass die Einstellung für manuelle Übersteuerung wahr ist, damit die manuellen Einstellungen für Drehzahl, Ausrichtung oder Anstellwinkel der Turbine verwendet werden können.

Manual override: On	Set: Yes
Turbine speed:	15 RPM

- Übersteuerung der Ausrichtung: Aktiviert die manuelle Übersteuerung der Turbinenausrichtung und die manuelle Einstellung der Ausrichtung. Das STEP 7-Anwenderprogramm verlangt, dass die Einstellung für manuelle Übersteuerung und für Übersteuerung der Ausrichtung wahr ist, damit die Ausrichtungseinstellung angewendet werden kann.

Yaw override: On	Set: Yes
Turbine yaw:	52 deg.

- Übersteuerung des Anstellwinkels: Aktiviert die manuelle Übersteuerung des Anstellwinkels der Rotorblätter. Das STEP 7-Anwenderprogramm verlangt, dass die Einstellung für manuelle Übersteuerung und für Übersteuerung des Anstellwinkels wahr ist, damit die Einstellung des Anstellwinkels der Rotorblätter angewendet werden kann.

Pitch override: On	Set: Yes
Blade pitch:	45 deg.

Die HTML-Seite enthält eine Schaltfläche "Senden", um die Übersteuerungseinstellungen an die Steuerung zu senden.

Submit override settings and values

Das Eingabefeld für die Bremsung bietet eine manuelle Einstellung eines Prozentwerts für die Bremsung. Das STEP 7-Anwenderprogramm verlangt keine manuelle Übersteuerung, um einen Bremswert zu übernehmen.

Braking:	25 %
----------	------

Außerdem nutzt die HTML-Seite einen AWP-Befehl, um die Sondervariable, die die Benutzer-ID des Benutzers enthält, der auf die Seite zugreift, in eine Variable in der PLC-Variablen-tabelle zu schreiben (Seite 559).

### 11.3.8.2 Steuerungsdaten lesen und anzeigen

Die HTML-Seite "Remote Wind Turbine Monitor" nutzt zahlreiche AWP-Befehle zum Lesen von Daten aus der Steuerung (Seite 554), um diese Daten auf der Seite anzuzeigen. Beachten Sie beispielsweise den HTML-Code zum Anzeigen der Leistungsausgabe in diesem Teil der Beispiel-Webseite:



#### Beispiel für den HTML-Code

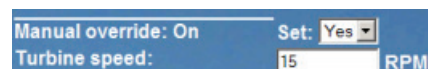
Der folgende Auszug aus der HTML-Seite "Remote Wind Turbine Monitor" zeigt den Text "Power Output:" in der linken Zelle einer Tabellenzeile an und liest die Variable für die Leistungsabgabe und zeigt sie in der rechten Zelle der Tabellenzeile zusammen mit dem Text für die Einheiten, KW, an.

Der AWP-Befehl := "Data\_block\_1".PowerOutput: führt die Leseoperation durch. Beachten Sie, dass Datenbausteine über den Namen und nicht über die Nummer des Datenbausteins referenziert werden (d.h. über "Data\_block\_1" und nicht über "DB1").

```
<tr style="height:2%;">
<td>
<p>Power output:</p>
</td>
<td>
<p style="margin-bottom:5px;"> := "Data_block_1".PowerOutput: KW</p>
</td>
</tr>
```

### 11.3.8.3 Enum-Typ verwenden

Die HTML-Seite "Remote Wind Turbine Monitor" verwendet Enum-Typen für die drei Stellen, an denen die HTML-Seite "ON" oder "OFF" für einen Booleschen Wert anzeigt und an der der Benutzer einen Booleschen Wert eingibt. Der Enum-Typ für "ON" führt zu einem Wert von 1, und der Enum-Typ für "OFF" führt zu einem Wert von 0. Beispiel: Betrachten Sie den HTML-Code zum Lesen und Schreiben der Einstellung zum Aktivieren der manuellen Übersteuerung im Wert "Data\_block\_1".ManualOverrideEnable mittels eines Enum-Typs:



### Beispiel für den HTML-Code

Die folgenden Auszüge aus der HTML-Seite "Remote Wind Turbine Monitor" zeigen die Deklaration eines Enum-Typs mit dem Namen "OverrideStatus" mit den Werten für "Off" und "On" von 0 und 1 und das nachfolgende Festlegen einer Enum-Typreferenz von OverrideStatus für die Boolesche Variable ManualOverrideEnable im Datenblock "Data\_block\_1". Beachten Sie, dass es für die Datenbausteinvariable oder die PLC-Variable für jede Deklaration AWP\_Enum\_Ref eine entsprechende Deklaration AWP\_In\_Variable geben muss, wenn die HTML-Seite über einen Enum-Typ in die Variable schreibt.

```
<!-- AWP_In_Variable Name='Data_block_1'.ManualOverrideEnable' -->
```

```
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
```

```
<!-- AWP_Enum_Ref Name='Data_block_1'.ManualOverrideEnable'
Enum="OverrideStatus" -->
```

Dabei beinhaltet die HTML-Seite ein Anzeigefeld in einer Tabellenzelle für den aktuellen Zustand von ManualOverrideEnable. Es wird lediglich ein normaler Lesebefehl für Variablen verwendet, doch dank des zuvor deklarierten und referenzierten Enum-Typs zeigt die Seite "Off" oder "On" und nicht 0 oder 1 an.

```
<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
```

```
<p>Manual override: :=Data_block_1.ManualOverrideEnable:</p>
```

```
</td>
```

Die HTML-Seite enthält eine Klappliste, damit der Anwender den Wert von ManualOverrideEnable ändern kann. Die Auswahlliste zeigt den Text "Yes" und "No" als Optionen an. Durch den Enum-Typ ist der Text "Yes" mit dem Wert "On" und der Text "No" ist dem Wert "Off" verbunden. Wird keine Auswahl getroffen, bleibt der Wert von ManualOverrideEnable unverändert.

```
<select name='Data_block_1'.ManualOverrideEnable'>
<option value=':Data_block_1.ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option selected value="Off">No</option>
</select>
```

Die Auswahlliste ist in einem Formular auf der HTML-Seite enthalten. Wenn der Anwender auf die Schaltfläche "Senden" klickt, lädt die Seite das Formular hoch. Dadurch wird der Wert "1" in die Boolesche Variable ManualOverrideEnable in Data\_block\_1 geschrieben, sofern der Anwender "Yes" ausgewählt hat, bzw. es wird "0" geschrieben, sofern der Anwender "No" ausgewählt hat.

#### 11.3.8.4 Benutzereingaben in die Steuerung schreiben

Die HTML-Seite "Remote Wind Turbine Monitor" umfasst verschiedene AWP-Befehle zum Schreiben von Daten in die Steuerung (Seite 555). Die HTML-Seite deklariert AWP\_In\_Variables für Boolesche Variablen, so dass ein Admin-Benutzer die Windturbine manuell steuern kann und die manuelle Übersteuerung für die Turbinendrehzahl, die Übersteuerung der Turbinenausrichtung und/oder des Anstellwinkels der Rotorblätter aktivieren kann. Die Seite umfasst zudem AWP\_In\_Variables, damit ein admin-Benutzer nachfolgend Gleitpunktwerte für Turbinendrehzahl, Ausrichtung, Anstellwinkel und Prozentwert für die Bremsung festlegen kann. Die Seite nutzt einen HTTP-Befehl zum Einstellen eines Formulars (Form POST), um die AWP\_In\_Variables in die Steuerung zu schreiben.



Beachten Sie z. B. den HTML-Code für die manuelle Einstellung des Bremswerts:

Braking:	<input type="text" value="25"/>	%
----------	---------------------------------	---

### Beispiel für den HTML-Code

Der folgende Ausschnitt aus der HTML-Seite "Remote Wind Turbine Monitor" deklariert zunächst eine AWP\_In\_Variable für "Data\_block\_1", die es der HTML-Seite ermöglicht, in beliebige Variablen im Datenbaustein "Data\_block\_1" zu schreiben. Die Seite zeigt den Text "Braking:" in der linken Zelle der Tabellenzeile an. In der rechten Zelle der Tabellenzeile befindet sich das Feld, in dem Benutzereingaben für die Variable "Braking" von "Data\_block\_1" möglich sind. Dieser Benutzereingabewert befindet sich in einem HTML-Formular, das die eingegebenen Textdaten über die HTTP-Methode "POST" in die CPU schreibt. Die Seite liest dann den tatsächlichen Verzögerungswert aus der Steuerung aus und zeigt ihn im Dateneingabefeld an.

Ein admin-Benutzer kann anschließend über diese Seite einen Bremswert in den Datenbaustein der CPU schreiben, der die Bremsung steuert.

```
<!-- AWP_In_Variable Name='Data_block_1' -->
...
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Braking:</p></td>
<td>
<form method="POST">
<p><input name='Data_block_1'.Braking' size="10" type="text"> %</p>
</form>
</td>
</tr>
```

---

#### Hinweis

Beachten Sie Folgendes: Wenn eine benutzerdefinierte Seite ein Dateneingabefeld für eine schreibbare Datenbausteinvariable vom Datentyp "String" enthält, muss der Anwender die Zeichenkette bei der Eingabe des Textes in das Feld in einfache Anführungszeichen setzen.

---

#### Hinweis

Beachten Sie, dass, wenn Sie einen gesamten Datenbaustein in einer Deklaration AWP\_In\_Variable deklarieren, z. B. <!-- AWP\_In\_Variable Name=""Datenbaustein\_1" -->, dann kann jede Variable in dem Datenbaustein über die benutzerdefinierte Webseite geschrieben werden. Gehen Sie auf diese Weise vor, wenn Sie möchten, dass alle Variablen in einem Datenbaustein schreibbar sein sollen. Andernfalls, wenn Sie möchten, dass nur spezifische Datenbausteinvariablen über die benutzerdefinierte Webseite beschreibbar sein sollen, deklarieren Sie dies spezifisch anhand einer Deklaration wie <!-- AWP\_In\_Variable Name=""Datenbaustein\_1".Bremsung' -->

---

### 11.3.8.5 Sondervariablen schreiben

Die Webseite "Remote Wind Turbine Monitor" schreibt die Sondervariable SERVER:current\_user\_id in eine PLC-Variablen in der CPU. In diesem Fall enthält der PLC-Variablenwert die Benutzer-ID des Benutzers, der auf die Webseite "Remote Wind Turbine Monitor" zugreift. Derzeit hat der admin-Benutzer die Benutzer-ID 1, deshalb ist der PLC-Variablenwert auf 1 gesetzt.

Die Sondervariable wird von der Webseite in das PLC-Gerät geschrieben und benötigt keine Bedienoberfläche.

### Beispiel für den HTML-Code

```
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
```

### 11.3.8.6 Referenz: HTML-Code der Webseite "Remote Wind Turbine Monitor"

#### Wind\_turbine.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><!--
Dieses Testprogramm simuliert eine Webseite zur Überwachung und
Bedienung einer Windturbine.
Erforderliche PLC-Variablen und Datenbausteinvariablen in STEP 7:
```

```
PLC-Variable:
User_ID: Int
```

```
Datenbausteine:
Data_block_1
```

```
Variablen in Data_Block_1:
Location: String
TurbineNumber: Int
WindSpeed: Real
WindDirection: Real
Temperature: Real
PowerOutput: Real
ManualOverrideEnable: Bool
TurbineSpeed: Real
YawOverride: Bool
Yaw: Real
PitchOverride: Bool
Pitch: Real
Braking: Real
```

Die benutzerdefinierte Webseite zeigt aktuelle Werte für die PLC-Daten und bietet eine Auswahlliste, um die drei Booleschen Werte mit zugewiesenem Aufzählungstyp festzulegen. Über die Schaltfläche "Senden" werden die ausgewählten Booleschen Werte ebenso wie die Dateneingabefelder für Drehzahl, Ausrichtung und Anstellwinkel der Turbine hochgeladen. Der Bremswert kann ohne die Schaltfläche "Senden" festgelegt werden.

Für die Verwendung dieser Seite ist kein tatsächliches STEP 7-Programm erforderlich. Theoretisch würde das STEP 7-Programm nur auf die Werte für Drehzahl, Ausrichtung und Anstellwinkel der Turbine reagieren, wenn die zugewiesenen Booleschen Werte festgelegt wären. Die einzige Anforderung an STEP 7 ist, die WWW-Anweisung mit der DB-Nummer der generierten Datenbausteine für diese Seite aufzurufen.

```
-->
<!-- AWP_In_Variable Name='Data_block_1' -->
<!-- AWP_In_Variable Name='Data_block_1'.ManualOverrideEnable' -->
<!-- AWP_In_Variable Name='Data_block_1'.PitchOverride' -->
<!-- AWP_In_Variable Name='Data_block_1'.YawOverride' -->
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
<!-- AWP_Enum_Ref Name='Data_block_1'.ManualOverrideEnable'
Enum="OverrideStatus" -->
<!-- AWP_Enum_Ref Name='Data_block_1'.PitchOverride'
Enum="OverrideStatus" -->
<!-- AWP_Enum_Ref Name='Data_block_1'.YawOverride'
Enum="OverrideStatus" -->

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"><link rel="stylesheet" href="Wind_turbine.css">
<title>Fernüberwachung von Windturbinen</title>
</head>
<body>
<table style="background-image: url('./Wind_turbine.jpg'); width: 904px; height: 534px;" cellpadding="0" cellspacing="2"><tr style="height: 2%;"><td colspan="2">
<h2>Remote Wind Turbine Monitor: Turbine
#:="Data_block_1".TurbineNumber: :="Data_block_1".Location:</h2>
</td>

<tr style="height: 2%;"><td style="width: 24%;"><p>Wind speed:</p></td>
<td><p> :="Data_block_1".WindSpeed: km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 24%;"><p>Wind direction:</p></td>
<td><p> :="Data_block_1".WindDirection: deg.</p></td>
</tr>

<tr style="height: 2%;"><td style="width: 24%;"><p>Temperature:</p></td>
<td><p> :="Data_block_1".Temperature: deg. C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 24%;"><p>Power output:</p></td>
<td style="margin-bottom:5px;"> :="Data_block_1".PowerOutput: KW</p>
</td>
</tr>
```

```
<form method="POST" action="">
<tr style="height: 2%;" >
<td style="width=24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: := "Data_block_1".ManualOverrideEnable:</p>
</td>
<td class="Text">Set:

<select name=' "Data_block_1".ManualOverrideEnable'>
<option value=' := "Data_block_1".ManualOverrideEnable:' > </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
24%;"><p>Turbine speed:</p></td>
<td>
<p style="margin-bottom:5px;"><input
name= "Data_block_1".TurbineSpeed' size="10"
value=' := "Data_block_1".TurbineSpeed:' type="text"> RPM</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Yaw override: := "Data_block_1".YawOverride: </p>
</td>
<td class="Text">Set:

<select name=' "Data_block_1".YawOverride'>
<option value=' := "Data_block_1".YawOverride:' > </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Turbine yaw:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name=' "Data_block_1".Yaw'
size="10" value=' := "Data_block_1".Yaw:' type="text"> deg.</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Pitch override: := "Data_block_1".PitchOverride: </p>
```

```

</td>
<td class="Text">Set:

<select name=' "Data_block_1".PitchOverride'>
<option value=':="Data_block_1".PitchOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width=24%; border-bottom-style: Solid; border-bottom-
width: 2px; border-bottom-color: #ffffff;">
<p>Blade pitch:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name=' "Data_block_1".Pitch'
size="10" value=':="Data_block_1".Pitch:' type="text"> deg.</p>
</td>

</tr>
<tr style="height: 2%;">
<td colspan="2"><br>
<input type="submit" value="Submit override settings and values">
</td>
</tr>
</form>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;"><p>Braking:</p></td>
<td>
<form method="POST" action="">
<p> <input name=' "Data_block_1".Braking' size="10"
value=':="Data_block_1".Braking:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>

```

## Wind\_turbine.css

```

H2 {
font-family: Arial;
font-weight: bold;
font-size: 14.0pt;
color: #FFFFFF;
margin-top: 0px;
margin-bottom: 10px;
}

```

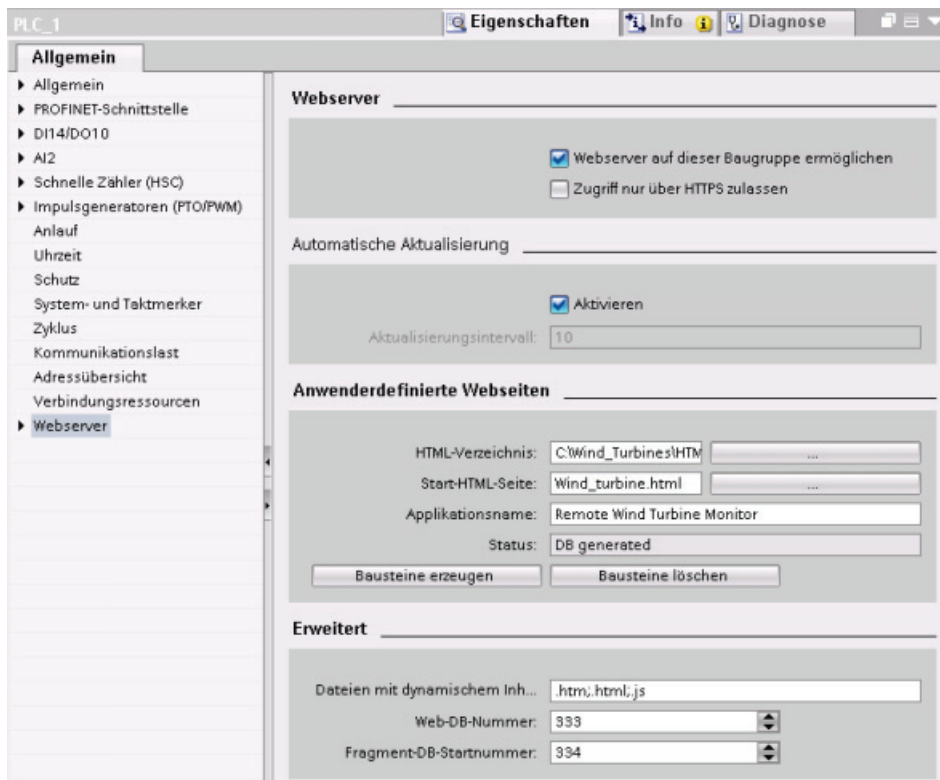
```

P {
  font-family: Arial;
  font-weight: bold;
  color: #FFFFFF;
  font-size: 12.0pt;
  margin-top:0px;
  margin-bottom:0px;
}
TD.Text {
  font-family: Arial;
  font-weight: bold;
  color: #FFFFFF;
  font-size: 12.0pt;
  margin-top:0px;
  margin-bottom:0px;
}
    
```

### 11.3.8.7 Konfiguration der Beispiel-Webseite in STEP 7

Um die HTML-Seite "Remote Wind Turbine Monitor" als benutzerdefinierte Webseite für die S7-1200 einzufügen, konfigurieren Sie die Daten für die HTML-Seite in STEP 7 und erstellen aus der HTML-Seite Datenbausteine.

Sie greifen auf die CPU-Eigenschaften der S7-1200, die die Windturbine steuert, zu und geben die Konfigurationsinformationen in die Eigenschaften der benutzerdefinierten Webseiten des Webservers ein:



## Konfigurationsfelder

- **HTML-Verzeichnis:** Dieses Feld gibt den vollständig qualifizierten Pfadnamen auf den Ordner an, in dem sich die Standardseite (Startseite) auf dem Computer befindet. Über die Schaltfläche "..." können Sie zu dem gewünschten Ordner blättern.
- **Standard-HTML-Seite:** Dieses Feld gibt den Dateinamen der Standardseite bzw. der Startseite der HTML-Anwendung an. Über die Schaltfläche "..." können Sie die gewünschte Datei auswählen. In diesem Beispiel ist WindTurbine.html die Standard-HTML-Seite. Das Beispiel "Remote Wind Turbine Monitor" besteht lediglich aus einer einzigen Seite, doch in anderen benutzerdefinierten Anwendungen können von der Standardseite weitere Seiten über Links aufgerufen werden. Im HTML-Code muss die Standardseite andere Seiten relativ zum HTML-Quellordner referenzieren.
- **Anwendungsname:** Dieses Feld ist optional und enthält den Namen, den der Webbrowser im Adressfeld anzeigt, wenn die Seite aufgerufen wird. In diesem Beispiel lautet der Name "Remote Wind Turbine Monitor", es kann jedoch ein beliebiger Name eingegeben werden.

Es müssen keine weiteren Felder konfiguriert werden.

## Abschließende Schritte

Um die HTML-Seite "Remote Wind Turbine Monitor" wie konfiguriert nutzen zu können, erstellen Sie die Bausteine, programmieren die WWW-Anweisung (Seite 569) mit der Nummer des generierten Steuer-DBs als Eingangsparameter, laden die Programmbausteine in die CPU und versetzen die CPU in den Betriebszustand RUN.

Wenn ein Bediener danach die Standard-Webseiten für die S7-1200 aufruft, von der die Windturbine gesteuert wird, kann er über den Link "Benutzerseiten" in der Navigationsleiste die Webseite "Remote Wind Turbine Monitor" anzeigen. Diese Seite bietet nun die Mittel, um die Windturbine zu überwachen und zu steuern.

### 11.3.9 Benutzerdefinierte Webseiten in mehreren Sprachen einrichten

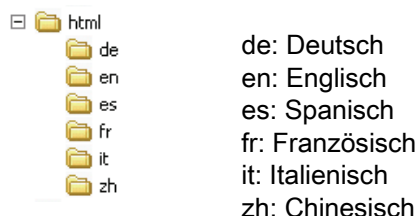
Der Webserver bietet Ihnen die Möglichkeit, benutzerdefinierte Webseiten in den folgenden Sprachen anzulegen:

- Deutsch (de)
- Englisch (en)
- Spanisch (es)
- Französisch (fr)
- Italienisch (it)
- Chinesisch (zh)

Hierfür legen Sie Ihre HTML-Seiten in einer Ordnerstruktur (Seite 584) entsprechend den Sprachen an und richten das spezifische Cookie "siemens\_automation\_language" für Ihre Seiten ein (Seite 584). Der Webserver reagiert auf dieses Cookie und schaltet zu der Standardseite in dem jeweiligen Sprachordner um.

### 11.3.9.1 Ordnerstruktur anlegen

Um benutzerdefinierte Webseiten in mehreren Sprachen bereitzustellen, richten Sie in Ihrem HTML-Verzeichnis eine Ordnerstruktur ein. Die aus zwei Buchstaben bestehenden Ordnernamen sind spezifisch und müssen wie im Folgenden gezeigt vergeben werden:



Auf der gleichen Ebene können Sie auch andere Ordner für Ihre Seiten anordnen, z. B. Ordner für Bilder oder Skripte.

Sie können beliebige Sprachordner einfügen. Sie brauchen nicht alle sechs Sprachen aufzunehmen. In den Sprachordnern erstellen und programmieren Sie Ihre HTML-Seiten in der jeweiligen Sprache.

### 11.3.9.2 Sprachumschaltung programmieren

Der Webserver führt die Sprachumschaltung anhand des Cookies "siemens\_automation\_language" durch. Dieses Cookie wird in den HTML-Seiten definiert und eingerichtet und vom Webserver ausgewertet, um eine Seite in der jeweiligen Sprache aus dem Sprachordner mit dem Namen der entsprechenden Sprache anzuzeigen. Die HTML-Seite muss JavaScript enthalten, um für dieses Cookie eine der vordefinierten Sprachkennungen einzurichten: "de", "en", "es", "fr", "it" oder "zh".

Setzt die HTML-Seite z. B. das Cookie auf "de", schaltet der Webserver zum Ordner "de" um und zeigt die Seite mit dem HTML-Standardseitenamen gemäß der STEP 7-Konfiguration (Seite 587) an.

#### Beispiel

Im folgenden Beispiel gibt es eine HTML-Standardseite mit dem Namen "langswitch.html" in jedem der Sprachordner. Im HTML-Verzeichnis befindet sich außerdem der Ordner "script". Der Ordner "script" enthält eine JavaScript-Datei mit dem Namen "lang.js". Jede Seite "langswitch.html" nutzt dieses JavaScript, um das Sprachcookie "siemens\_automation\_language" festzulegen.

#### HTML für "langswitch.html" im Ordner "en"

Die Kopfzeile der HTML-Seite richtet die Sprache "Englisch" ein, legt den Zeichensatz "UTF-8" fest und gibt den Pfad der JavaScript-Datei "lang.js" an.



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Language switching english page</title>
<script type="text/javascript" src="script/lang.js" ></script>

```

Der Inhalt der Datei enthält eine Auswahlliste, über die der Anwender zwischen Deutsch und Englisch wählen kann. Englisch ("en") ist die vorausgewählte Sprache. Wenn der Anwender die Sprache wechselt, ruft die Seite die JavaScript-Funktion DoLocalLanguageChange() mit dem Wert der ausgewählten Option auf.

```

<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top" nowrap>
      <!-- change language immediately on selection change -->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        size="1">
        <option value="de" >German</option>
        <option value="en" selected >English</option>
      </select>
    </td>
  </tr>
</table><!-- Language Selection End-->

```

### HTML für "langswitch.html" im Ordner "de"

Die Kopfzeile für die deutsche Seite "langswitch.html" ist mit der Kopfzeile der englischen Seite identisch, außer dass die deutsche Sprache eingestellt ist.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="de"><meta http-
equiv="Content-Type" content="text/html; charset=utf-8">
<title>Sprachumschaltung Deutsche Seite</title>
<script type="text/javascript" src="script/lang.js" ></script>
</head>

```

Der HTML-Code auf der deutschen Seite ist mit dem der englischen Seite identisch, außer dass der Standardwert der ausgewählten Sprache Deutsch ("de") ist.

```

<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top" nowrap>
      <!-- change language immediately on change of the selection -
->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        <size="1">
          <option value="de" selected >Deutsch</option>
          <option value="en" >Englisch</option>
        </select>
      </td>
    </tr>
  </table><!-- Language Selection End-->

```

### JavaScript "lang.js" im Ordner "script"

Die Funktion "DoLocalLanguageChange()" befindet sich in der Datei "lang.js". Diese Funktion ruft die Funktion "SetLangCookie()" auf und aktualisiert dann das Fenster, in dem die HTML-Seite angezeigt wird.

Die Funktion "SetLangCookie()" weist dem Cookie "siemens\_automation\_language" des Dokuments den Wert aus der Auswahlliste zu. Die Funktion legt auch den Pfad der Anwendung fest, so dass die umgeschaltete Seite und nicht die anfordernde Seite den Wert des Cookies empfängt.

Optional kann die Seite im kommentierten Abschnitt einen Wert für die Gültigkeit des Cookies angeben.

```

function DoLocalLanguageChange(oSelect) {
    SetLangCookie(oSelect.value);
    top.window.location.reload();
}
function SetLangCookie(value) {
    var strval = "siemens_automation_language=";
    // Dies ist das Cookie, über das der Websserver
    // die gewünschte Sprache erkennt.
    // Dieser Name ist für den Websserver erforderlich.
    strval = strval + value;
    strval = strval + "; path=/ ";
    // Geben Sie den Pfad für die Anwendung an, weil ansonsten
    // der Pfad auf die anfordernde Seite gelegt werden würde
    // und diese Seite das Cookie nicht erhalten würde.
    /* OPTIONAL
    Geben Sie eine Ablaufzeit an, wenn dieses Cookie länger
existieren soll
als die aktuelle Browser-Sitzung:
    var now = new Date();
    var endttime = new Date(now.getTime() + expiration);
    strval = strval + "; expires=" +
        endttime.toGMTString() + ";";
    */
    document.cookie = strval;
}

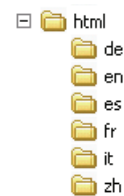
```

### 11.3.9.3 STEP 7 für die Verwendung einer mehrsprachigen Seitenstruktur konfigurieren

Um mehrsprachige benutzerdefinierte Webseiten zu konfigurieren, gehen Sie ähnlich vor wie beim Konfigurieren von benutzerdefinierten Webseiten (Seite 567). Wenn Sie jedoch Ordner für Sprachen eingerichtet haben, geben Sie in der Einstellung des HTML-Verzeichnisses den Ordner an, der die einzelnen Sprachordner enthält. Sie richten das HTML-Verzeichnis nicht als einen der Sprachordner ein.

Wenn Sie die HTML-Standardseite auswählen, navigieren Sie in den Sprachordner und wählen die HTML-Seite aus, die die Startseite sein soll. Wenn Sie anschließend Bausteine generieren und die Bausteine in die CPU laden, zeigt der Webserver die Startseite aus dem konfigurierten Sprachordner an.

Beispiel: Befindet sich die hier gezeigte Ordnerstruktur auf C:\, ist die Einstellung für das HTML-Verzeichnis C:\html. Und wenn zunächst die englische Seite angezeigt werden soll, navigieren Sie für die Einstellung der HTML-Standardseite zum Pfad en\langswitch.html.



### 11.3.10 Erweiterte Steuerung von benutzerdefinierten Webseiten

Wenn Sie für Ihre benutzerdefinierten Webseiten Datenbausteine generieren, erstellt STEP 7 einen Steuer-DB, der die Anzeige von und die Interaktion mit den benutzerdefinierten Webseiten steuert. STEP 7 erstellt zudem einen Satz DB-Fragmente, die die einzelnen Seiten darstellen. Unter normalen Umständen müssen Sie die Struktur des Steuer-DBs nicht kennen und brauchen auch nicht zu wissen, wie Sie ihn ändern.

Wenn Sie beispielsweise eine Webanwendung ein- und ausschalten möchten oder einzelne Fragmente manuell ändern möchten, können Sie dies über die Variablen des Steuer-DBs und die WWW-Anweisung erreichen.

#### Struktur des Steuer-DBs

Der Steuer-DB besitzt eine umfangreiche Datenstruktur und ist zugänglich, wenn Sie Ihr STEP 7-Anwenderprogramm programmieren. Hier werden nur einige der Variablen des Steuerdatenbausteins beschrieben.

#### Commandstate-Struktur

"Commandstate" ist eine Struktur, die globale Befehle und globale Zustände für den Webserver enthält.

##### Globale Befehle in der "Commandstate"-Struktur

Die globalen Befehle gelten für den Webserver im Allgemeinen. Sie können den Webserver über die Parameter des Steuer-DBs deaktivieren oder neu starten.

Bausteinvariable	Datentyp	Beschreibung
init	BOOL	Auswertung des Steuer-DB und Initialisierung der Webanwendung
deactivate	BOOL	Deaktivierung der Webanwendung

**Globale Zustände in der Commandstate-Struktur**

Die globalen Zustände gelten für den Webserver im Allgemeinen und enthalten Statusinformationen über die Webanwendung.

Bausteinvariable	Datentyp	Beschreibung
initializing	BOOL	Webanwendung liest den Steuer-DB
error	BOOL	Webanwendung konnte nicht initialisiert werden
deactivating	BOOL	Webanwendung wird beendet
deactivated	BOOL	Webanwendung ist beendet
initialized	BOOL	Webanwendung ist initialisiert

**Anforderungstabelle**

Die Anforderungstabelle ist ein Array aus Strukturen mit Befehlen und Zuständen, die für einzelne DB-Fragmente gelten. Wenn Sie mit dem Befehl AWP\_Start\_Fragment (Seite 563) Fragmente des "manuellen" Typs angelegt haben, muss das STEP 7-Anwenderprogramm diese Seiten über den Steuer-DB steuern. Die Anforderungszustände sind schreibgeschützt und liefern Informationen zum aktuellen Fragment. Die Anforderungsbefehle nutzen Sie, um das aktuelle Fragment zu steuern.

Bausteinvariable	Datentyp	Beschreibung
requesttab	ARRAY [ 1 .. 4 ] OF STRUCT	Array aus Strukturen für die Steuerung einzelner DB-Fragmente Der Webserver kann bis zu vier Fragmente gleichzeitig bearbeiten. Der Array-Index für ein bestimmtes Fragment ist beliebig, wenn der Webserver mehrere Fragmente oder Fragmente aus mehreren Browser-Sessions verarbeitet.

**Struct-Elemente der Requesttab-Struktur**

Bausteinvariable	Datentyp	Beschreibung
page_index	UINT	Nummer der aktuellen Webseite
fragment_index	UINT	Nummer des aktuellen Fragments - Festlegung eines anderen Fragments möglich
// Anforderungsbefehle		

Bausteinvariable	Datentyp	Beschreibung
continue	BOOL	Gibt die aktuelle Seite bzw. das aktuelle Fragment zum Senden frei und fährt mit dem nächsten Fragment fort
repeat	BOOL	Gibt die aktuelle Seite bzw. das aktuelle Fragment zum erneuten Senden frei und fährt mit demselben Fragment fort
abort	BOOL	HTTP-Verbindung ohne Senden schließen
finish	BOOL	Dieses Fragment senden; Seite ist vollständig - keine weiteren Fragmente bearbeiten
// Anforderungszustände		Die Anforderungszustände sind schreibgeschützt.
idle	BOOL	Nichts durchzuführen, jedoch aktiv
waiting	BOOL	Fragment wartet auf Freigabe
sending	BOOL	Fragment sendet
aborting	BOOL	Benutzer hat die aktuelle Anforderung abgebrochen

## Funktionsweise

Immer wenn Ihr Programm den Steuer-DB ändert, muss es die WWW-Anweisung mit der Nummer des geänderten Steuer-DBs als Parameter aufrufen. Die globalen Befehle und Anforderungsbefehle werden wirksam, wenn das STEP 7-Anwenderprogramm die WWW-Anweisung (Seite 569) ausführt.

Das STEP 7-Anwenderprogramm kann `fragment_index` explizit festlegen und bewirkt dadurch, dass der Webserver das angegebene Fragment mit einem Anforderungsbefehl verarbeitet. Andernfalls verarbeitet der Webserver das aktuelle Fragment für die aktuelle Seite, wenn die WWW-Anweisung ausgeführt wird.

Mögliche Techniken zur Verwendung von `fragment_index` sind u.a.:

- Aktuelles Fragment verarbeiten: `fragment_index` unverändert lassen und den Fortsetzungsbefehl setzen.
- Aktuelles Fragment überspringen: `fragment_index` auf 0 setzen und den Fortsetzungsbefehl setzen
- Aktuelles Fragment durch ein anderes Fragment ersetzen: Für `fragment_index` die neue Fragment-ID angeben und den Fortsetzungsbefehl setzen

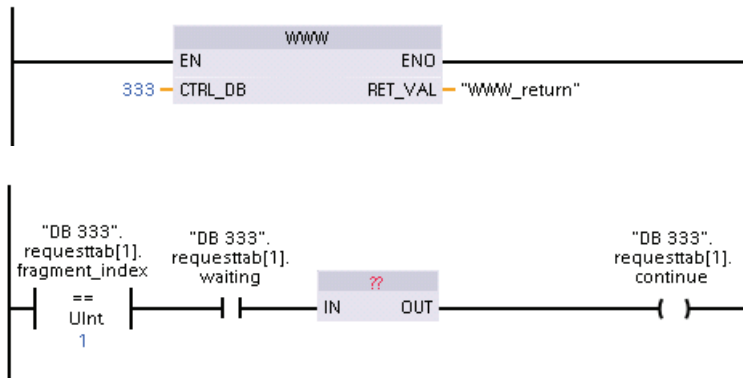
Um sich möglicherweise verändernde globale Zustände oder Anforderungszustände abzufragen, muss das STEP 7-Anwenderprogramm die WWW-Anweisung aufrufen, um die aktuellen Werte dieser Zustände auszuwerten. Eine typische Verwendung könnte es sein, die WWW-Anweisung regelmäßig aufzurufen, bis ein bestimmter Zustand eintritt.

**Hinweis**

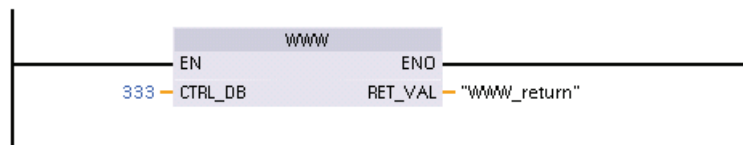
Wenn das STEP 7-Anwenderprogramm mehr als einen Anforderungsbefehl setzt, verarbeitet die WWW-Anweisung nur einen davon in dieser Reihenfolge: abbrechen, beenden, wiederholen, fortfahren. Die WWW-Anweisung löscht alle Anforderungsbefehle nach der Verarbeitung.

**Beispiele**

Das folgende Beispiel zeigt ein STEP 7-Anwenderprogramm, das auf ein Fragment mit der ID 1 im Wartezustand prüft, nachdem zuvor die WWW-Anweisung aufgerufen wurde. Es wartet möglicherweise auch, dass andere anwendungsspezifische Bedingungen auftreten. Dann führt es die für das Fragment erforderliche Verarbeitung durch, wobei es z. B. Datenbausteinvariablen setzt, Berechnungen durchführt oder andere anwendungsspezifische Aufgaben verarbeitet. Danach setzt es den Fortsetzungsmerker, damit der Webserver dieses Fragment ausführt.



Wenn das Programm die WWW-Anweisung mit diesem geänderten Steuer-DB aufruft, kann die benutzerdefinierte Webseite mit diesem Fragment vom Webbrowser angezeigt werden.



Beachten Sie, dass es sich hier um ein vereinfachtes Beispiel handelt. Das zu prüfende Fragment kann sich in jeder der vier requesttab-Strukturen des Arrays befinden.

## 11.4 Einschränkungen

Die folgenden IT-Aspekte können sich auf Ihre Nutzung des Webserver auswirken:

- Üblicherweise müssen Sie die IP-Adresse der CPU eingeben, um auf die Standard-Webseiten bzw. die benutzerdefinierten Webseiten zuzugreifen. Wenn Ihr Webbrowser keine direkte Verbindung mit einer IP-Adresse gestattet, wenden Sie sich an Ihren IT-Administrator. Wenn Ihre lokalen Richtlinien DNS unterstützen, können Sie über einen DNS-Eintrag eine Verbindung zu der IP-Adresse herstellen.
- Firewalls, Proxy-Einstellungen und andere standortspezifische Einschränkungen können ebenfalls den Zugriff auf die CPU begrenzen. Um solche Probleme zu beheben, wenden Sie sich an Ihren IT-Administrator.
- Die Standard-Webseiten verwenden JavaScripts und Cookies. Wenn JavaScripts oder Cookies in Ihrem Webbrowser deaktiviert sind, aktivieren Sie sie. Wenn Sie sie nicht aktivieren können, sind einige Funktionen eingeschränkt (Seite 592). Die Verwendung von JavaScripts und Cookies in benutzerdefinierten Webseiten ist optional. Wenn Sie sie verwenden, müssen sie in Ihrem Browser aktiviert sein.
- Der Webserver unterstützt Secure Sockets Layer (SSL). Sie können die Standard-Webseiten und benutzerdefinierten Webseiten über eine der beiden URLs `http://ww.xx.yy.zz` oder `https://ww.xx.yy.zz` aufrufen. Dabei steht "ww.xx.yy.zz" für die IP-Adresse der CPU.
- Siemens bietet für den sicheren Zugriff auf den Webserver ein Sicherheitszertifikat. Auf der Standard-Webseite "Einführung" (Seite 537) können Sie das Zertifikat herunterladen und in die Internetoptionen Ihres Webbrowsers importieren (Seite 593). Wenn Sie das Zertifikat nicht importieren, wird Ihnen bei jedem Aufruf des Webserver über die `https://`-Adresse eine Sicherheitsabfrage angezeigt.

### Anzahl der Verbindungen

Der Webserver unterstützt maximal 30 aktive HTTP-Verbindungen. Diese 30 Verbindungen können auf verschiedene Weise genutzt werden, je nach dem von Ihnen verwendeten Webbrowser und der Anzahl der unterschiedlichen Objekte pro Seite (CSS-Dateien, Bilder, weitere HTML-Dateien). Einige Verbindungen bleiben bestehen, während die Seite angezeigt wird, andere werden nach der Erstverbindung freigegeben.

Wenn Sie beispielsweise den Browser Mozilla Firefox 8 nutzen, der maximal sechs beständige Verbindungen unterstützt, könnten Sie fünf Browser oder Browserregister verwenden, bevor der Webserver beginnt, Verbindungen abzubrechen. Sollte eine Seite nicht alle sechs Verbindungen nutzen, können Sie weitere Browser oder Browserregister verwenden.

Beachten Sie auch, dass die Performance der Seiten durch die Anzahl der aktiven Verbindungen beeinflusst werden kann.

### 11.4.1 Eingeschränkte Funktionen bei deaktiviertem JavaScript

Die Standard-Webseiten sind über HTML, JavaScripts und Cookies implementiert. Wenn an Ihrem Standort die Verwendung von JavaScripts und Cookies eingeschränkt ist, aktivieren Sie diese, damit die Webseiten einwandfrei funktionieren. Wenn Sie für Ihren Webbrowser keine JavaScripts aktivieren können, können die von JavaScripts gesteuerten Funktionen nicht ausgeführt werden.

#### Allgemein

Die Seiten werden nicht dynamisch aktualisiert. Sie müssen die Seiten manuell über das Symbol zum Aktualisieren (Seite 535) aktualisieren, um die aktuellen Daten anzuzeigen.

#### Diagnosepufferseite

- Einzelheiten zu Ereignissen anzeigen: Mit JavaScript wählen Sie eine Zeile im Diagnosepuffer aus, um im unteren Abschnitt Einzelheiten dazu anzuzeigen. Ohne JavaScript müssen Sie auf den Hyperlink des Ereignisfelds eines Diagnosepuffereintrags klicken, um die Ereignisdaten im unteren Bereich anzuzeigen.
- Bereich der anzuzeigenden Diagnosepuffereinträge ändern: Mit JavaScript können Sie über die Klappliste oben auf der Seite den Bereich der anzuzeigenden Diagnosepuffereinträge auswählen, woraufhin die Seite automatisch aktualisiert wird. Ohne JavaScript können Sie über die Klappliste oben auf der Seite den Bereich der anzuzeigenden Diagnosepuffereinträge auswählen, doch Sie müssen dann auf den Link "Go" klicken, um die Diagnosepufferseite entsprechend zu aktualisieren.

Beachten Sie, dass der Hyperlink "Go" und die Hyperlinks der Ereignisfelder nur angezeigt werden, wenn JavaScript nicht aktiviert ist. Diese Links sind nicht erforderlich und deshalb auch nicht vorhanden, wenn JavaScript aktiviert ist.

---

#### Hinweis

Der Browser Opera V11.0 unterstützt nicht die Schaltfläche "Go" und auch keine Diagnoseeinträge in Form von Hyperlinks. Bei Opera V11.0 können Sie, wenn Sie JavaScript deaktiviert haben, keine Ereignisdetails aufrufen und den Bereich nicht ändern.

---

#### Modulinformationsseite

- Sie können keine Daten filtern.
- Sie können keine Felder sortieren.

#### Variablenzustandsseite

- Nachdem Sie eine Variable eingegeben haben, müssen Sie den Fokus manuell in die Zeile "New variable" setzen, um eine neue Variable einzugeben.
- Wenn Sie ein Anzeigeformat auswählen, wird der Datenwert nicht automatisch im ausgewählten Format angezeigt. Sie müssen auf die Schaltfläche "Monitor value" klicken, um die Anzeige mit dem neuen Format zu aktualisieren.



### Datenprotokollseite

- Sie können nicht auf einen Dateinamen unter "Recent entries" klicken, um eine Protokolldatei zu öffnen oder zu speichern. Sie können für die gleiche Funktionalität jedoch das Symbol zum Herunterladen nutzen.
- Die Datenprotokollseite wird nicht aktualisiert.
- Die Schaltflächen "+" und "-" zum Inkrementieren und Dekrementieren der Anzahl der letzten Einträge sind funktionslos.
- Die Eingabe eines Wertes direkt in die Anzahl der letzten Einträge bewirkt nicht, dass die Anzahl der Einträge neu festgelegt wird. Wenn Sie in Mozilla Firefox versuchen, einen Wert in dieses Feld einzugeben, erscheint eine leere Anzeige. Um die Anzeige der Datenprotokolle wiederherzustellen, müssen Sie im Navigationsfenster erneut "Data Logs" auswählen. Das Feld für die Anzahl der letzten Einträge wird nicht verändert.

Beachten Sie, dass Sie die Datenprotokollseite verlassen und wieder aufrufen können, um die neuesten 25 Einträge anzuzeigen.

### 11.4.2 Eingeschränkte Funktionen, wenn keine Cookies zugelassen sind

Wenn Ihr Webbrowser keine Cookies zulässt, können Sie sich nicht als Benutzer "admin" anmelden.

### 11.4.3 Siemens-Sicherheitszertifikat importieren

Sie können das Siemens-Sicherheitszertifikat in Ihre Internetoptionen importieren, damit Ihnen nicht bei jedem Aufruf von <https://ww.xx.yy.zz> (dabei ist "ww.xx.yy.zz" dies IP-Adresse der CPU) in Ihrem Webbrowser die Sicherheitabfrage angezeigt wird. Wenn Sie eine <http://>-Adresse und keine <https://>-URL eingeben, brauchen Sie das Zertifikat nicht herunterzuladen und zu installieren.

### Herunterladen des Zertifikats

Klicken Sie auf der Einführungsseite (Seite 537) auf den Link "Zertifikat herunterladen", um das Siemens-Sicherheitszertifikat auf Ihren PC herunterzuladen. Die Vorgehensweise unterscheidet sich dabei je nach Webbrowser:

[download certificate](#)

### Zertifikat im Internet Explorer importieren

1. Klicken Sie auf der Einführungsseite auf den Link "Zertifikat herunterladen". Es wird der Dialog "Dateidownload - Sicherheitswarnung" angezeigt.
2. Klicken Sie im Dialog "Dateidownload - Sicherheitswarnung" auf "Öffnen", um die Datei zu öffnen. Der Dialog "Zertifikat" wird angezeigt.

3. Klicken Sie im Dialog "Zertifikat" auf die Schaltfläche "Zertifikat installieren", um den Assistenten zum Importieren des Zertifikats aufzurufen.
4. Folgen Sie den Dialogen im "Zertifikatimport-Assistent", um das Zertifikat zu importieren. Dabei wählt das Betriebssystem automatisch den Zertifikatspeicher aus.

### Zertifikat in Mozilla Firefox importieren

1. Klicken Sie auf der Einführungsseite auf den Link "Zertifikat herunterladen". Der Dialog "MiniWebCA\_Cer.crt öffnen" wird angezeigt.
2. Klicken Sie im Dialog "MiniWebCA\_Cer.crt öffnen" auf "Datei speichern". Der Dialog "Downloads" wird angezeigt.
3. Doppelklicken Sie im Dialog "Downloads" auf "MiniWebCA\_Cer.crt". Wenn Sie den Download mehrmals gestartet haben, werden mehrere Kopien angezeigt. Doppelklicken Sie auf einen beliebigen Eintrag "MiniWebCA\_Cer.crt".
4. Klicken Sie auf "OK", wenn Sie bestätigen sollen, dass eine ausführbare Datei geöffnet wird.
5. Klicken Sie ggf. im Dialog "Datei öffnen - Sicherheitswarnung" auf "Öffnen". Der Dialog "Zertifikat" wird angezeigt.
6. Klicken Sie im Dialog "Zertifikat" auf die Schaltfläche "Zertifikat installieren".
7. Folgen Sie den Dialogen im "Zertifikatimport-Assistent", um das Zertifikat zu importieren. Dabei wählt das Betriebssystem automatisch den Zertifikatspeicher aus.
8. Wenn der Dialog "Sicherheitswarnung" angezeigt wird, klicken Sie auf "Ja", um die Installation des Zertifikats zu bestätigen.

### Andere Browser

Folgen Sie den Konventionen Ihres Webbrowsers, um das Siemens-Zertifikat zu importieren und zu installieren.

Nachdem Sie das Siemens-Sicherheitszertifikat "S7-Controller Family" in den Internetoptionen für die Inhalte Ihres Webbrowsers installiert haben, brauchen Sie keine Sicherheitsabfrage zu bestätigen, wenn Sie über [https:// ww.xx.yy.zz](https://ww.xx.yy.zz) auf den Webserver zugreifen.

#### 11.4.4 Datenprotokolle im CSV-Format in nicht amerikanische/englische Versionen von Microsoft Excel importieren

Die Protokolldateien sind im CSV-Dateiformat (durch Komma getrennte Werte) gespeichert. Sie können diese Dateien von der Datenprotokollseite direkt in Excel öffnen, wenn auf Ihrem System eine amerikanische oder englische Version von Excel installiert ist. In anderen Ländern ist dieses Format jedoch nicht so weit verbreitet, weil Kommas häufig in numerischer Notation auftreten.

Wenn Sie nicht eine amerikanische/englische Version von Excel verwenden, gehen Sie folgendermaßen vor, um eine gespeicherte Protokolldatei zu öffnen:

1. Öffnen Sie Microsoft Excel und legen Sie eine leere Arbeitsmappe an.
2. Wählen Sie im Menü "Daten > Externe Daten importieren" den Befehl "Daten importieren".
3. Navigieren Sie zu der Protokolldatei, die Sie öffnen möchten, und wählen Sie sie aus. Der Textimport-Assistent wird aufgerufen.
4. Ändern Sie im Textimport-Assistenten die Standardoption für "Ursprünglicher Datentyp" von "Feste Breite" in "Getrennt".
5. Wählen Sie die Schaltfläche "Weiter".
6. Aktivieren Sie im Dialog "Schritt 2" das Kontrollkästchen "Komma", um den Trennzeichentyp von "Tab" in "Komma" zu ändern.
7. Wählen Sie die Schaltfläche "Weiter".
8. Im Dialog "Schritt 3" können Sie optional das Datenformat von MDJ (Monat/Tag/Jahr) in ein anderes Format ändern.
9. Führen Sie die übrigen Schritte im Textimport-Assistenten durch, um die Datei zu importieren.



## Kommunikationsprozessor

### 12.1 Mit den seriellen Kommunikationsschnittstellen arbeiten

Zwei Kommunikationsmodule (CMs) und ein Kommunikationsboard (CB) bieten die Schnittstelle für die PtP-Kommunikation:

- CM 1241 RS232 (Seite 872)
- CM 1241 RS422/485 (Seite 873)
- CB 1241 RS485 (Seite 869)

Sie können bis zu drei CMs (jeden Typs) plus ein CB, insgesamt vier Kommunikationsschnittstellen anschließen. Bauen Sie das CM links von der CPU oder eines anderen CMs ein. Bauen Sie das CB auf der Vorderseite der CPU ein. Ausführliche Anweisungen zum Ein- und Ausbau des Moduls finden Sie im Kapitel "Einbau" im S7-1200 Systemhandbuch (Seite 59).

Die seriellen Kommunikationsschnittstellen haben die folgenden Eigenschaften:

- Potentialgetrennter Anschluss
- Unterstützung von Punkt-zu-Punkt-Protokollen
- Konfiguration und Programmierung über erweiterte Anweisungen und Bibliotheksfunktionen
- Anzeige der Sende- und Empfangsaktivität über LEDs
- Diagnose-LED (nur CMs)
- Spannungsversorgung über die CPU: Keine externe Stromversorgung erforderlich.

Siehe technische Daten der Kommunikationsschnittstellen (Seite 861).

#### LED-Anzeigen

Die Kommunikationsmodule haben drei LED-Anzeigen:

- Diagnose-LED (DIAG): Diese LED blinkt rot, bis sie von der CPU angesprochen wird. Nach dem Anlauf der CPU prüft diese auf CMs und adressiert diese. Die Diagnose-LED beginnt, grün zu blinken. Das bedeutet, dass die CPU das CM adressiert, ihm jedoch noch keine Konfiguration zugewiesen hat. Die CPU lädt die Konfiguration in die konfigurierten CMs, wenn das Programm in die CPU geladen wird. Nach dem Laden in die CPU muss die Diagnose-LED am Kommunikationsmodul dauerhaft grün leuchten.
- Sende-LED (Tx): Die Sende-LED leuchtet, wenn Daten über den Kommunikationsport gesendet werden.
- Empfangs-LED (Rx): Diese LED leuchtet, wenn Daten über den Kommunikationsport empfangen werden.

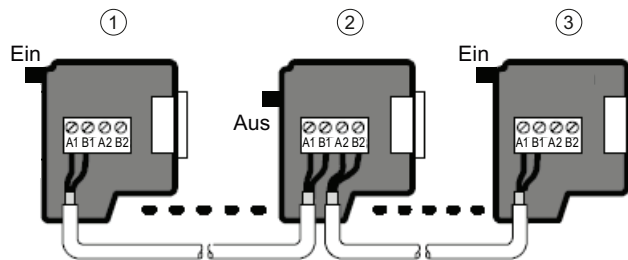
Das Kommunikationsboard bietet LEDs für Senden (TxD) und Empfangen (RxD). Es hat keine Diagnose-LED.

## 12.2 Abschließen eines RS485-Busanschlussteckers

Siemens bietet einen RS485-Busanschlusstecker (Seite 885), mit dem Sie problemlos mehrere Geräte an ein RS485-Netz anschließen können. Der Stecker verfügt über zwei Klemmensätze, mit denen Sie die Eingangs- und Ausgangskabel für das Netz befestigen können. Der Stecker verfügt zudem über Schalter, um das Netz selektiv abschließen zu können.

### Hinweis

Sie schließen nur die zwei Enden des RS485-Netzes ab. Die Geräte zwischen den beiden Endgeräten werden nicht abgeschlossen. Blanker Kabelschirm: Ca. 12 mm muss blank auf der Metallführung liegen.



- ① Schalterstellung = Ein: Abschlusswiderstand zugeschaltet
- ② Schalterstellung = Aus: Abschlusswiderstand nicht zugeschaltet
- ③ Schalterstellung = Ein: Abschlusswiderstand zugeschaltet

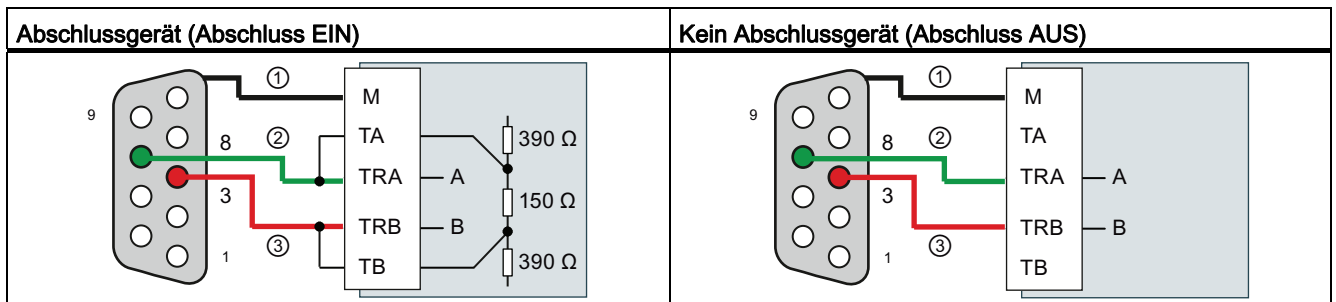
Tabelle 12- 1 Abschluss des RS485-Steckers

Abschlussgerät (Abschluss EIN)	Kein Abschlussgerät (Abschluss AUS)

- ① Pinnummer
- ② Busanschlusstecker
- ③ Kabelschirm

Das CB 1241 bietet interne Widerstände für den Netzabschluss. Um die Verbindung abzuschließen, verbinden Sie TRA mit TA und TRB mit TB, um die internen Widerstände in den Schaltkreis aufzunehmen. CB 1241 hat keinen 9-poligen Steckverbinder. Die folgende Tabelle zeigt die Anschlüsse an einen 9-poligen Steckverbinder am Kommunikationspartner.

Tabelle 12- 2 Abschluss des CB 1241



① M an den Kabelschirm anschließen

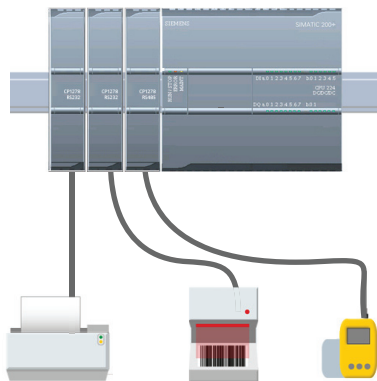
② A = TxD/RxD - (grüne Ader/Pol 8)

③ B = TxD/RxD + (rote Ader/Pol 3)

## 12.3 Punkt-zu-Punkt-Kommunikation (PtP)

Die CPU unterstützt die folgende Punkt-zu-Punkt-Kommunikation (PtP) für zeichenbasierte serielle Protokolle. Das Punkt-zu-Punkt-Protokoll bietet maximale Freiheit und Flexibilität, erfordert jedoch eine aufwändige Implementierung im Anwenderprogramm.

- PtP (Seite 599)
- USS (Seite 642)
- Modbus (Seite 659)



PtP bietet eine Vielzahl an Möglichkeiten:

- Direktes Senden von Informationen an ein externes Gerät, wie z. B. einen Drucker
- Empfangen von Informationen von anderen Geräten, wie z. B. Strichcodelesern, RFID-Lesern, Kamera- oder Überwachungssystemen anderer Hersteller und vielen anderen Gerätearten
- Informationsaustausch, Senden und Empfangen von Daten mit anderen Geräten, wie z. B. GPS-Geräten, Kamera- oder Überwachungssystemen anderer Hersteller, Funkmodems und vielen anderen

Die PtP-Kommunikation ist eine serielle Kommunikation mit Standard-UARTs, die verschiedene Baudraten und Paritäten unterstützen. Die RS232- und RS422/485-Kommunikationsmodule sowie das RS485-Kommunikationsboard bilden die elektrischen Schnittstellen für die Durchführung der PtP-Kommunikation.

### 12.3.1 Punkt-zu-Punkt-Operationen

#### 12.3.1.1 Gemeinsame Parameter für Punkt-zu-Punkt-Operationen

Tabelle 12- 3 Allgemeine Eingangsparameter der PTP-Anweisungen

Parameter	Beschreibung
REQ	<p>Viele der PtP-Anweisungen haben einen Eingang REQ, der die Anweisung bei einer steigenden Flanke (0 nach 1) initiiert. Der Eingang REQ muss während der Ausführung einer Anweisung 1 (WAHR) sein, doch der Eingang REQ kann so lange wie gewünscht WAHR bleiben. Die Anweisung stößt erst dann eine andere Anweisung an, wenn sie mit Eingang REQ = FALSCH aufgerufen wird, so dass die Anweisung den Zustand des Eingangs REQ zurücksetzen kann. Dies ist erforderlich, damit die Anweisung die steigende Flanke zum Starten der nächsten Anweisung erkennen kann.</p> <p>Wenn Sie eine PtP-Anweisung in Ihr Programm einfügen, werden Sie von STEP 7 aufgefordert, den Instanz-DB anzugeben. Verwenden Sie für jeden PtP-Anweisungsaufwurf einen eindeutigen DB. Dadurch wird sichergestellt, dass jede Anweisung die Eingänge wie REQ ordnungsgemäß verarbeitet.</p>
PORT	<p>Eine Portadresse wird während der Konfiguration des Kommunikationsgeräts zugewiesen. Nach der Konfiguration kann aus der Klappliste des Parameters ein symbolischer Name für den Standardport ausgewählt werden. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Konstanten" der PLC-Variablen-tabelle zugewiesen.</p>
Auflösung von Bitzeiten	<p>Für verschiedene Parameter wird die Anzahl der Bitzeiten bei der konfigurierten Baudrate angegeben. Durch Angabe des Parameters in Bitzeiten ist der Parameter unabhängig von der Baudrate. Alle Parameter mit Einheiten von Bitzeiten können mit einer maximalen Anzahl von 65535 angegeben werden. Der maximale Zeitraum jedoch, der von einem CM oder CB gemessen werden kann, beträgt 8 Sekunden.</p>

Die Ausgangsparameter DONE, NDR, ERROR und STATUS der PtP-Anweisungen zeigen den Ausführungsstatus der PtP-Funktionen an.

Tabelle 12- 4 Ausgangsparameter DONE, NDR, ERROR und STATUS

Parameter	Datentyp	Standard	Beschreibung
DONE	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehler abgeschlossen wurde; andernfalls FALSCH.
NDR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die angeforderte Aktion fehlerfrei abgeschlossen und neue Daten empfangen wurden; andernfalls FALSCH.



Parameter	Datentyp	Standard	Beschreibung
ERROR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehlern abgeschlossen wurde, der entsprechende Fehlercode befindet sich in STATUS; andernfalls FALSCH.
STATUS	Word	0	<p>Ergebniszustand:</p> <ul style="list-style-type: none"> <li>• Wird das Bit DONE oder NDR gesetzt, wird STATUS auf 0 oder auf einen Informationscode gesetzt.</li> <li>• Wird das Bit ERROR gesetzt, wird STATUS auf einen Fehlercode gesetzt.</li> <li>• Wird keines der oben aufgeführten Bits gesetzt, kann die Anweisung Statusergebnisse zurückgeben, die den aktuellen Zustand der Funktion beschreiben.</li> </ul> <p>Der Wert in STATUS wird während der Ausführung der Funktion gehalten.</p>

#### Hinweis

Beachten Sie, dass die Parameter DONE, NDR und ERROR nur eine Ausführung lang gesetzt sind. Ihre Programmlogik muss temporäre Ausgangszustandswerte in Datenpuffern speichern, damit Sie in nachfolgenden Programmzyklen Zustandsänderungen erkennen können.

Tabelle 12- 5 Gemeinsame BedingungsCodes

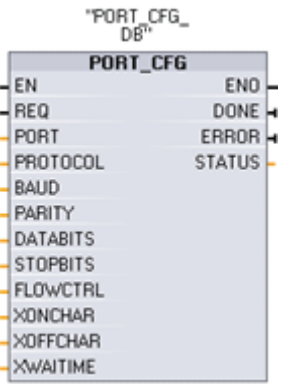
STATUS (W#16#...)	Beschreibung
0000	Kein Fehler
7000	Funktion nicht besetzt
7001	Funktion ist beim ersten Aufruf aktiv.
7002	Funktion ist bei Folgeaufrufen aktiv (Abfragen nach dem ersten Aufruf).
8x3A	Unzulässiger Pointer in Parameter x
8070	Gesamter interner Speicher ausgelastet, zu viele Anweisungen gleichzeitig in Bearbeitung
8080	Portnummer ist unzulässig.
8081	Zeitüberschreitung, Modulfehler oder anderer interner Fehler
8082	Parametrierung fehlgeschlagen, weil Parametrierung im Hintergrund läuft.
8083	Pufferüberlauf: Das CM oder CB hat eine Empfangsmeldung mit einer Länge, die größer ist als vom Längenparameter zugelassen, zurückgegeben.
8090	Interner Fehler: Falsche Meldungslänge, falsches Submodul oder unzulässige Meldung Wenden Sie sich an den Kunden-Support.
8091	Interner Fehler: Falsche Version in Parametrierungsmeldung Wenden Sie sich an den Kunden-Support.
8092	Interner Fehler: Falsche Datensatzlänge in Parametrierungsmeldung Wenden Sie sich an den Kunden-Support.

Tabelle 12- 6 Gemeinsame Fehlerklassen

Beschreibung der Klasse	Fehlerklassen	Beschreibung
Portkonfiguration	80Ax	Für die Beschreibung häufiger Fehler in der Schnittstellenkonfiguration
Sendekonfiguration	80Bx	Für die Beschreibung häufiger Fehler in der Sendekonfiguration
Empfangskonfiguration	80Cx	Für die Beschreibung häufiger Fehler in der Empfangskonfiguration
Sende-Laufzeit	80Dx	Für die Beschreibung häufiger Laufzeitfehler beim Senden
Empfangs-Laufzeit	80Ex	Für die Beschreibung häufiger Laufzeitfehler beim Empfangen
Signalverarbeitung	80Fx	Für die Beschreibung häufiger Fehler in Verbindung mit der Signalverarbeitung

### 12.3.1.2 Operation PORT\_CFG

Tabelle 12- 7 Anweisung PORT\_CFG (Port-Konfiguration)

KOP/FUP	SCL	Beschreibung
 <p>The screenshot shows a ladder logic diagram for the PORT_CFG DB. It includes a network with a coil labeled 'PORT_CFG' and several normally open contacts labeled EN, REQ, PORT, PROTOCOL, BAUD, PARITY, DATABITS, STOPBITS, FLOWCTRL, XONCHAR, XOFFCHAR, and XWAITIME. On the right side, there are output coils labeled END, DONE, ERROR, and STATUS.</p>	<pre>"PORT_CFG_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   PROTOCOL:=_uint_in_,   BAUD:=_uint_in_,   PARITY:=_uint_in_,   DATABITS:=_uint_in_,   STOPBITS:=_uint_in_,   FLOWCTRL:=_uint_in_,   XONCHAR:=_char_in_,   XOFFCHAR:=_char_in_,   WAITTIME:=_uint_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>Mit PORT_CFG können Sie Portparameter wie die Baudrate über Ihr Programm ändern.</p> <p>Sie können die statische Anfangskonfiguration des Ports in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit PORT_CFG in Ihrem Programm die Konfiguration ändern.</p>

<sup>1</sup> STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Die Konfigurationsänderungen von PORT\_CFG werden in der CPU nicht dauerhaft gespeichert. Die in der Gerätekonfiguration gespeicherten Parameter werden nicht wiederhergestellt, wenn die CPU von RUN in STOP wechselt oder neu eingeschaltet wird. Weitere Informationen finden Sie unter Kommunikationsports konfigurieren (Seite 618) und Flusskontrolle verwalten (Seite 620).

Tabelle 12- 8 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke dieses Eingangs. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
PROTOCOL	IN	UInt	0 - Protokoll der Punkt-zu-Punkt-Kommunikation (Standardwert) 1..n - künftige Definition spezifischer Protokolle
BAUD	IN	UInt	Baudrate des Ports (Standardwert: 0): 1 = 300 Baud, 2 = 600 Baud, 3 = 1200 Baud, 4 = 2400 Baud, 5 = 4800 Baud, 6 = 9600 Baud, 7 = 19200 Baud, 8 = 38400 Baud, 9 = 57600 Baud, 10 = 76800 Baud, 11 = 115200 Baud
PARITY	IN	UInt	Parität des Ports (Standardwert: 0): 1 = Keine Parität, 2 = Gerade Parität, 3 = Ungerade Parität, 4 = Mark-Parität, 5 = Space-Parität
DATABITS	IN	UInt	Bits pro Zeichen (Standardwert): 1 = 8 Datenbits, 2 = 7 Datenbits
STOPBITS	IN	UInt	Stoppbits (Standardwert: 0): 1 = 1 Stoppbit, 2 = 2 Stoppbits
FLOWCTRL	IN	UInt	Flusskontrolle (Standardwert: 0): 1 = Keine Flusskontrolle, 2 = XON/XOFF, 3 = Hardware-RTS immer EIN, 4 = Hardware-RTS geschaltet
XONCHAR	IN	Char	Gibt das Zeichen an, das als XON-Zeichen dient. Dies ist typischerweise ein DC1-Zeichen (11H). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 0)
XOFFCHAR	IN	Char	Gibt das Zeichen an, das als XOFF-Zeichen dient. Dies ist typischerweise ein DC3-Zeichen (13H). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 0)
XWAITIME	IN	UInt	Gibt an, wie lange nach dem Empfang eines XON-Zeichens auf ein XOFF-Zeichen gewartet werden soll bzw. wie lange nach Aktivierung von RTS auf das CTS-Signal gewartet werden soll (0 bis 65535 ms). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 2000)
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle 12- 9 Bedingungs-codes

STATUS (W#16#....)	Beschreibung
80A0	Angegebenes Protokoll ist nicht vorhanden.
80A1	Angegebene Baudrate ist nicht vorhanden.

STATUS (W#16#...)	Beschreibung
80A2	Angegebene Paritätsoption ist nicht vorhanden.
80A3	Angegebene Anzahl Datenbits ist nicht vorhanden.
80A4	Angegebene Anzahl Stoppbits ist nicht vorhanden.
80A5	Angegebene Art der Flusskontrolle ist nicht vorhanden.
80A6	Wartezeit ist 0 und Flusskontrolle ist aktiviert
80A7	XON und XOFF sind unzulässige Werte (z. B. der gleiche Wert)

### 12.3.1.3 Operation SEND\_CFG

Tabelle 12- 10 Anweisung SEND\_CFG (Sendekonfiguration)

KOP/FUP	SCL	Beschreibung
	<pre>"SEND_CFG_DB" (     REQ:=_bool_in_,     PORT:=_uint_in_,     RTSONDLY:=_uint_in_,     RTSOFFDLY:=_uint_in_,     BREAK:=_uint_in_,     IDLELINE:=_uint_in_,     DONE=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	<p>SEND_CFG ermöglicht das dynamische Konfigurieren serieller Übertragungsparameter für eine PtP-Kommunikationsschnittstelle. In einem CM oder CB anstehende Meldungen werden gelöscht, wenn SEND_CFG ausgeführt wird.</p>

1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Sie können die statische Anfangskonfiguration des Ports in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit SEND\_CFG in Ihrem Programm die Konfiguration ändern.

Die Konfigurationsänderungen von SEND\_CFG werden in der CPU nicht dauerhaft gespeichert. Die in der Gerätekonfiguration gespeicherten Parameter werden nicht wiederhergestellt, wenn die CPU von RUN in STOP wechselt oder neu eingeschaltet wird. Siehe Sendeparameter konfigurieren (Seite 622).

Tabelle 12- 11 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
PORT	IN	PORT

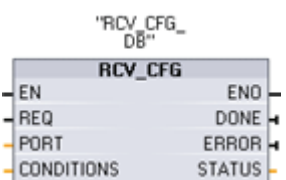
Parameter und Datentyp	Datentyp	Beschreibung	
RTSONDLY	IN	UInt	Anzahl der Millisekunden, die nach Aktivierung von RTS gewartet werden soll, bevor eine Übertragung von Tx-Daten durchgeführt wird. Dieser Parameter ist nur gültig, wenn die Hardwareflusskontrolle aktiviert ist. Der gültige Bereich ist 0 bis 65535 ms. Der Wert 0 deaktiviert die Funktion. (Standardwert: 0)
RTSOFFDLY	IN	UInt	Anzahl der Millisekunden, die nach der Übertragung von Tx-Daten gewartet werden soll, bevor RTS deaktiviert wird: Dieser Parameter ist nur gültig, wenn die Hardwareflusskontrolle aktiviert ist. Der gültige Bereich ist 0 bis 65535 ms. Der Wert 0 deaktiviert die Funktion. (Standardwert: 0)
BREAK	IN	UInt	Dieser Parameter gibt an, dass beim Start jeder Meldung für die angegebene Anzahl von Bitzeiten eine Pause gesendet wird. Maximal sind 65535 Bitzeiten bis maximal 8 Sekunden möglich. Der Wert 0 deaktiviert die Funktion. (Standardwert: 12)
IDLELINE	IN	UInt	Dieser Parameter gibt an, dass vor dem Start jeder Meldung für die angegebene Anzahl von Bitzeiten die Leitung im Leerlauf bleibt. Maximal sind 65535 Bitzeiten bis maximal 8 Sekunden möglich. Der Wert 0 deaktiviert die Funktion. (Standardwert: 12)
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle 12- 12 Bedingungscode

STATUS (W#16#....)	Beschreibung
80B0	Konfiguration eines Sendealarms ist nicht zulässig. Wenden Sie sich an den Kunden-Support.
80B1	Pausenzeit ist größer als der maximal zulässige Wert.
80B2	Leerlaufzeit ist größer als der maximal zulässige Wert.

### 12.3.1.4 Operation RCV\_CFG

Tabelle 12- 13 Anweisung RCV\_CFG (Empfangskonfiguration)

KOP/FUP	SCL	Beschreibung
	<pre>"RCV_CFG_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   CONDITIONS:=_struct_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	RCV_CFG führt die dynamische Konfiguration serieller Empfangsparameter für eine PtP-Kommunikationsschnittstelle durch. Diese Anweisung konfiguriert die Bedingungen, die den Beginn und das Ende einer empfangenen Meldung kennzeichnen. In einem CM oder CB anstehende Meldungen werden gelöscht, wenn RCV_CFG ausgeführt wird.

<sup>1</sup> STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Sie können die statische Anfangskonfiguration des Kommunikationsanschlusses in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit RCV\_CFG in Ihrem Programm die Konfiguration ändern.

Die Konfigurationsänderungen von RCV\_CFG werden in der CPU nicht dauerhaft gespeichert. Die in der Gerätekonfiguration gespeicherten Parameter werden nicht wiederhergestellt, wenn die CPU von RUN in STOP wechselt oder neu eingeschaltet wird. Weitere Informationen finden Sie unter Empfangsparameter konfigurieren (Seite 622).

Tabelle 12- 14 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke an diesem Eingang. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
CONDITIONS	IN	CONDITIONS	Die Datenstruktur von CONDITIONS gibt die Anfangs- und Endbedingungen der Meldung wie im Folgenden beschrieben an.
DONE	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

### Anfangsbedingungen für die Anweisung RCV\_PTP

Die Anweisung RCV\_PTP nutzt die von der Anweisung RCV\_CFG angegebene Konfiguration, um Anfang und Ende von Meldungen der Punkt-zu-Punkt-Kommunikation zu ermitteln. Der Meldungsbeginn wird von den Startbedingungen festgelegt. Der Meldungsbeginn kann anhand von einer oder mehreren Startbedingungen ermittelt werden. Sind mehrere Startbedingungen angegeben, müssen alle Bedingungen erfüllt sein, damit die Meldung gestartet wird.

Eine Beschreibung der Bedingungen für den Meldungsbeginn finden Sie unter "Empfangsparameter konfigurieren (Seite 623)".

## Datentypstruktur des Parameters CONDITIONS, Teil 1 (Startbedingungen)

Tabelle 12- 15 Struktur von CONDITIONS für Startbedingungen

Parameter und Datentyp		Datentyp	Beschreibung
STARTCOND	IN	UInt	Angabe der Startbedingung (Standardwert: 1) <ul style="list-style-type: none"> <li>• 01H - Startzeichen</li> <li>• 02H - Beliebiges Zeichen</li> <li>• 04H - Leitungspause</li> <li>• 08H - Leitung im Leerlauf</li> <li>• 10H - Zeichenfolge 1</li> <li>• 20H - Zeichenfolge 2</li> <li>• 40H - Zeichenfolge 3</li> <li>• 80H - Zeichenfolge 4</li> </ul>
IDLETIME	IN	UInt	Die Anzahl der erforderlichen Bitzeiten für Timeout des Leitungsleerlaufs. (Standardwert: 40). Nur in Verbindung mit der Bedingung "Leitung im Leerlauf". 0 bis 65535
STARTCHAR	IN	Byte	Das Startzeichen für die Bedingung "Startzeichen". (Standardwert: B#16#2)
STRSEQ1CTL	IN	Byte	Zeichenfolge 1, Steuerung für jedes Zeichen ignorieren/vergleichen: (Standardwert: B#16#0) Dies sind die Aktivierungsbits für jedes Zeichen der Startzeichenfolge. <ul style="list-style-type: none"> <li>• 01H - Zeichen 1</li> <li>• 02H - Zeichen 2</li> <li>• 04H - Zeichen 3</li> <li>• 08H - Zeichen 4</li> <li>• 10H - Zeichen 5</li> </ul> Wird ein Bit für ein bestimmtes Zeichen deaktiviert, bedeutet dies, dass an dieser Position der Zeichenfolge jedes Zeichen eine Übereinstimmung darstellt.
STRSEQ1	IN	Char[5]	Zeichenfolge 1, Startzeichen (5 Zeichen). Standardwert: 0
STRSEQ2CTL	IN	Byte	Zeichenfolge 2, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0)
STRSEQ2	IN	Char[5]	Zeichenfolge 2, Startzeichen (5 Zeichen). Standardwert: 0
STRSEQ3CTL	IN	Byte	Zeichenfolge 3, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0
STRSEQ3	IN	Char[5]	Zeichenfolge 3, Startzeichen (5 Zeichen). Standardwert: 0
STRSEQ4CTL	IN	Byte	Zeichenfolge 4, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0
STRSEQ4	IN	Char[5]	Zeichenfolge 4, Startzeichen (5 Zeichen), Standardwert: 0

**Beispiel**

Sehen Sie sich die folgende empfangene hexadezimal-codierte Meldung an: "68 10 aa 68 bb 10 aa 16". Die konfigurierten Startzeichenfolgen finden Sie in der folgenden Tabelle. Startzeichenfolgen werden ausgewertet, nachdem das erste Zeichen 68H erfolgreich empfangen wurde. Nach erfolgreichem Empfang des vierten Zeichens (das zweite 68H) ist die Startbedingung 1 erfüllt. Wenn die Startbedingungen erfüllt sind, beginnt die Auswertung der Endbedingungen.

Die Verarbeitung der Startzeichenfolge kann aufgrund verschiedener Fehler bei Parität, Framing oder Zeitabständen zwischen den Zeichen beendet werden. Diese Fehler führen dazu, dass die Meldung nicht empfangen wird, weil die Startbedingung nicht erfüllt wurde.

Tabelle 12- 16 Startbedingungen

Startbedingung	Erstes Zeichen	Erstes Zeichen +1	Erstes Zeichen +2	Erstes Zeichen +3	Erstes Zeichen +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

**Endbedingungen für die Anweisung RCV\_PTP**

Das Ende einer Meldung wird von den angegebenen Endbedingungen festgelegt. Das Ende einer Meldung wird durch das erste Auftreten einer oder mehrerer konfigurierter Endbedingungen festgelegt. Im Abschnitt "Bedingungen für den Meldungsbeginn" unter "Empfangsparameter konfigurieren (Seite 623)" werden die Endbedingungen beschrieben, die Sie für die Anweisung RCV\_CFG konfigurieren können.

Sie können die Endbedingungen entweder in den Eigenschaften der Kommunikationsschnittstelle in der Gerätekonfiguration oder über die Anweisung RCV\_CFG konfigurieren. Immer wenn die CPU von STOP in RUN wechselt, werden die Empfangsparameter (Start- und Endbedingungen) wieder auf die Einstellungen der Gerätekonfiguration gesetzt. Wenn das STEP 7 Anwenderprogramm RCV\_CFG ausführt, werden die Einstellungen auf die Bedingungen von RCV\_CFG gesetzt.



## Datentypstruktur des Parameters CONDITIONS, Teil 2 (Endebedingungen)

Tabelle 12- 17 Struktur von CONDITIONS für Endebedingungen

Parameter	Parametertyp	Datentyp	Beschreibung
ENDCOND	IN	UInt 0	Dieser Parameter gibt die Bedingung für das Meldungsende an: <ul style="list-style-type: none"> <li>• 01H - Antwortzeit</li> <li>• 02H - Meldungszeit</li> <li>• 04H - Zeichenabstand</li> <li>• 08H - Maximale Länge</li> <li>• 10H - N + LEN + M</li> <li>• 20H - Zeichenfolge</li> </ul>
MAXLEN	IN	UInt 1	Maximale Meldungslänge: Wird nur verwendet, wenn die Endebedingung "Maximale Länge" ausgewählt ist. 1 bis 1024 Bytes
N	IN	UInt 0	Byteposition des Längenfelds in der Meldung. Wird nur bei der Endebedingung N + LEN + M verwendet. 1 bis 1022 Bytes
LENGTHSIZE	IN	UInt 0	Größe des Bytefelds (1, 2 oder 4 Byte). Wird nur bei der Endebedingung N + LEN + M verwendet.
LENGTHM	IN	UInt 0	Geben Sie die Anzahl der Zeichen nach dem Längenfeld an, die nicht im Wert des Längenfelds enthalten sind. Diese Angabe wird nur bei der Endebedingung N + LEN + M verwendet. 0 bis 255 Bytes
RCVTIME	IN	UInt 200	Geben Sie an, wie lange auf das erste empfangene Zeichen gewartet werden soll. Die Empfangsanweisung wird mit einem Fehler beendet, wenn nicht innerhalb der angegebenen Zeit ein Zeichen erfolgreich empfangen wird. Diese Angabe wird nur bei der Bedingung "Antwortzeit" verwendet. (0 bis 65535 Bitzeiten, maximal 8 Sekunden)  Bei diesem Parameter handelt es sich nicht um eine Endebedingung, weil die Auswertung beendet wird, wenn das erste Zeichen einer Antwort empfangen wird. Es handelt sich nur in dem Sinn um eine Endebedingung, als dass eine Empfangsanweisung beendet wird, weil bei Erwartung einer Antwort keine Antwort empfangen wird. Sie müssen eine getrennte Endebedingung definieren.
MSGTIME	IN	UInt 200	Geben Sie an, wie lange nach dem Empfang des ersten Zeichens auf den vollständigen Empfang der gesamten Meldung gewartet werden soll. Dieser Parameter wird nur verwendet, wenn die Bedingung "Timeout der Meldung" ausgewählt ist. (0 bis 65535 ms)
CHARGAP	IN	UInt 12	Geben Sie die Anzahl der Bitzeiten zwischen den Zeichen an. Wenn die Anzahl der Bitzeiten zwischen den Zeichen den angegebenen Wert überschreitet, ist die Endebedingung erfüllt. Diese Angabe wird nur bei der Bedingung "Zeichenabstand" verwendet. (0 bis 65535 Bitzeiten, maximal 8 Sekunden)

12.3 Punkt-zu-Punkt-Kommunikation (PtP)

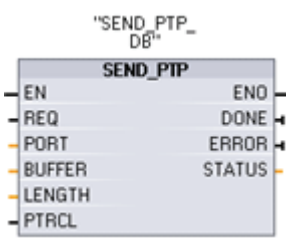
Parameter	Parametertyp	Datentyp	Beschreibung
ENDSEQ1CTL	IN	Byte B#16#0	Zeichenfolge 1, Steuerung für jedes Zeichen ignorieren/vergleichen:  Dies sind die Aktivierungsbits für jedes Zeichen der Endezeichenfolge. Zeichen 1 ist Bit 0, Zeichen 2 ist Bit 1, ..., Zeichen 5 ist Bit 4. Wird ein Bit für ein bestimmtes Zeichen deaktiviert, bedeutet dies, dass an dieser Position der Zeichenfolge jedes Zeichen eine Übereinstimmung darstellt.
ENDSEQ1	IN	Char[5] 0	Zeichenfolge 1, Startzeichen (5 Zeichen)

Tabelle 12- 18 Bedingungs-codes

STATUS (W#16#...)	Beschreibung
80C0	Unzulässige Startbedingung ausgewählt
80C1	Unzulässige Endbedingung ausgewählt, keine Endbedingung ausgewählt
80C2	Empfangsalarm aktiviert und dies ist nicht möglich.
80C3	Endbedingung "Maximale Länge" ist aktiviert und die maximale Länge ist 0 oder > 1024.
80C4	Berechnete Länge ist aktiviert und N ist >= 1023.
80C5	Berechnete Länge ist aktiviert und Länge ist nicht 1, 2 oder 4.
80C6	Berechnete Länge ist aktiviert und der Wert von M ist > 255.
80C7	Berechnete Länge ist aktiviert und die berechnete Länge ist > 1024.
80C8	Timeout der Antwort ist aktiviert und das Antwort-Timeout ist null.
80C9	Timeout für den Zeichenabstand ist aktiviert und das Timeout ist null.
80CA	Timeout für den Leitungleerlauf ist aktiviert und das Timeout ist null.
80CB	Endezeichenfolge ist aktiviert, doch alle Zeichen sind "nicht relevant".
80CC	Startzeichenfolge (eine von 4) ist aktiviert, doch alle Zeichen sind "nicht relevant".

12.3.1.5 Operation SEND\_PTP

Tabelle 12- 19 Anweisung SEND\_PTP (Punkt-zu-Punkt-Daten senden)

KOP/FUP	SCL	Beschreibung
	<pre>"SEND_PTP_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   BUFFER:=_variant_in_,   LENGTH:=_uint_in_,   PTRCL:=_bool_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>SEND_PTP startet die Übertragung der Daten und überträgt den zugewiesenen Puffer zur Kommunikationsschnittstelle. Das Programm der CPU wird weiterhin ausgeführt, während das CM oder CB die Daten mit der zugewiesenen Baudrate sendet. Es darf zu jeder Zeit nur eine Sendeanweisung anstehen. Das CM oder CB gibt einen Fehler aus, wenn eine zweite Anweisung SEND_PTP ausgeführt wird, während das CM oder CB bereits eine Meldung sendet.</p>

1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 12- 20 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Freigabe der angeforderten Übertragung bei steigender Flanke dieses Freigabeeingangs. Dadurch wird der Inhalt des Puffers zur Punkt-zu-Punkt-Kommunikationsschnittstelle übertragen. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
BUFFER	IN	Variante	Dieser Parameter zeigt auf die Anfangsadresse des Sendepuffers. (Standardwert: 0) <b>Hinweis:</b> Boolesche Daten oder Boolesche Felder werden nicht unterstützt.
LENGTH	IN	UInt	Übertragene Rahmenlänge in Byte (Standardwert: 0) Verwenden Sie beim Senden einer komplexen Struktur immer die Länge 0.
PTRCL	IN	Bool	Dieser Parameter wählt den Puffer für die normale Punkt-zu-Punkt-Kommunikation oder für spezifische Siemens-Protokolle aus, die im angeschlossenen CM oder CB implementiert sind. (Standardwert: Falsch) FALSE = vom Anwenderprogramm gesteuerte Punkt-zu-Punkt-Anweisungen. (einzige gültige Option)
DONE	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Wenn eine Sendeanweisung in Bearbeitung ist, sind die Ausgänge DONE und ERROR im Zustand FALSCH. Nach dem Ende einer Sendeanweisung wird einer der Ausgänge DONE oder ERROR auf WAHR gesetzt, um den Zustand der Sendeanweisung zu melden. Während DONE oder ERROR im Zustand WAHR ist, ist der Ausgang STATUS gültig.

Die Anweisung gibt den Status 16#7001 aus, wenn die Kommunikationsschnittstelle die Sendedaten annimmt. Nachfolgende Ausführungen von SEND\_PTP geben den Wert 16#7002 aus, wenn das CM oder CB immer noch sendet. Nach dem Ende der Sendeanweisung gibt das CM oder CB den Status 16#0000 für die Sendeanweisung aus (sofern kein Fehler aufgetreten ist). Nachfolgende Ausführungen von SEND\_PTP mit REQ = 0 geben den Status 16#7000 (frei) aus.

Das folgende Diagramm zeigt die Beziehung zwischen den Ausgangswerten und REQ. Hierbei wird vorausgesetzt, dass die Anweisung regelmäßig aufgerufen wird, um den Status des Sendevorgangs zu prüfen. In der folgenden Abbildung wird davon ausgegangen, dass die Anweisung in jedem Zyklus aufgerufen wird (dargestellt durch die STATUS-Werte).

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

Die folgende Abbildung zeigt, wie die Parameter DONE und STATUS nur einen Zyklus lang gültig sind, wenn an der REQ-Leitung (einen Zyklus lang) ein Impuls anliegt, um die Sendeanweisung anzustoßen.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

Die folgende Abbildung zeigt die Beziehung der Parameter DONE, ERROR und STATUS im Fehlerfall.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

Die Werte DONE, ERROR und STATUS sind nur solange gültig, bis SEND\_PTP erneut mit dem gleichen Instanz-DB ausgeführt wird.

Tabelle 12- 21 Bedingungscode

STATUS (W#16#....)	Beschreibung
80D0	Neue Anforderung bei aktivem Sender
80D1	Sendung abgebrochen, weil kein CTS innerhalb der Wartezeit
80D2	Sendung abgebrochen, weil kein DSR vom DCE-Gerät
80D3	Sendung wegen Überlauf der Warteschlange abgebrochen (mehr als 1024 Bytes senden)
80D5	Umgekehrtes Biassignal (Drahtbruch)
833A	Der DB für den Parameter BUFFER ist nicht vorhanden.

### Interaktion der Parameter LENGTH und BUFFER für SEND\_PTP

Die Mindestdatengröße, die von der Anweisung SEND\_PTP gesendet werden kann, ist ein Byte. Der Parameter BUFFER legt die Größe der zu sendenden Daten fest. Sie können für den Parameter BUFFER weder den Datentyp Bool noch Arrays vom Typ Bool verwenden.

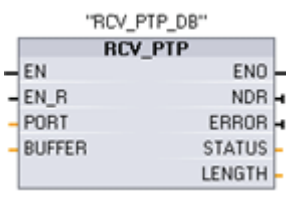
Sie können den Parameter LENGTH jederzeit auf 0 setzen und sicherstellen, dass SEND\_PTP die gesamte Datenstruktur des Parameters BUFFER sendet. Wenn Sie nur einen Teil der Datenstruktur des Parameters BUFFER senden möchten, können Sie LENGTH wie folgt setzen:

Tabelle 12- 22 Parameter LENGTH und BUFFER

LENGTH	BUFFER	Beschreibung
= 0	Nicht verwendet	Die vollständigen Daten werden wie vom Parameter BUFFER definiert gesendet. Wenn LENGTH = 0, brauchen Sie die Anzahl der übertragenen Bytes nicht anzugeben.
> 0	Elementarer Datentyp	Der LENGTH-Wert muss die Bytezahl dieses Datentyps enthalten. Beispiel: Bei einem Word-Wert muss LENGTH zwei sein. Bei einem Dword- oder Real-Wert muss LENGTH vier sein. Andernfalls erfolgt keine Übertragung, und der Fehler 8088H wird ausgegeben.
	Struktur	Der LENGTH-Wert kann eine Bytezahl kleiner als die vollständige Bytelänge der Struktur enthalten, dann werden nur die ersten LENGTH Bytes der Struktur aus BUFFER gesendet. Weil die interne Byteanordnung einer Struktur nicht immer bestimmt werden kann, kann dies zu unerwarteten Ergebnissen führen. Verwenden Sie dann eine LENGTH = 0, um die vollständige Struktur zu senden.
	Array	Der LENGTH-Wert muss eine Bytezahl kleiner als die vollständige Bytelänge des Arrays enthalten, wobei es sich um ein Vielfaches der Bytezahl des Datenelements handeln muss. Beispiel: Der Parameter LENGTH eines Arrays vom Typ Word muss ein Vielfaches von zwei sein und bei einem Array vom Typ Real ein Vielfaches von vier. Wenn LENGTH angegeben wird, wird die Anzahl der Array-Elemente in LENGTH-Bytes übertragen. Wenn beispielsweise BUFFER ein Array mit 15 Dword-Elementen (insgesamt 60 Bytes) enthält und Sie geben LENGTH = 20 an, werden die ersten fünf Dword-Elemente aus dem Array übertragen.  Der LENGTH-Wert muss ein Vielfaches der Bytezahl der Datenelemente sein. Andernfalls ist STATUS = 8088H, ERROR = 1 und keine Übertragung erfolgt.
	String	Der Parameter LENGTH enthält die Anzahl der zu sendenden Zeichen. Nur die Zeichen des String werden übertragen. Die Bytes mit der maximalen und der tatsächlichen Länge des String werden nicht gesendet.

### 12.3.1.6 Operation RCV\_PTP

Tabelle 12- 23 Anweisung RCV\_PTP (Punkt-zu-Punkt-Daten empfangen)

KOP/FUP	SCL	Beschreibung
	<pre>"RCV_PTP_DB" (     EN_R:=_bool_in_,     PORT:=_uint_in_,     BUFFER:=_variant_in_,     NDR=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_,     LENGTH=&gt;_uint_out_);</pre>	<p>RCV_PTP prüft die Meldungen, die im CM oder CB empfangen wurden. Wenn eine Meldung verfügbar ist, wird sie vom CM oder CB zur CPU übertragen. Ein Fehler gibt den entsprechenden STATUS-Wert aus.</p>

<sup>1</sup> STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 12- 24 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
EN_R	IN	Bool	Wenn dieser Eingang WAHR ist und eine Meldung verfügbar ist, wird die Meldung vom CM oder DB in BUFFER übertragen. Wenn EN_R = FALSCH ist, wird das CM oder CB auf Meldungen geprüft und die Ausgänge NDR, ERROR und STATUS werden aktualisiert, doch die Meldung wird nicht in BUFFER übertragen. (Standardwert: 0)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
BUFFER	IN	Variante	Dieser Parameter zeigt auf die Anfangsadresse des Empfangspuffers. Dieser Puffer muss groß genug sein, um die maximale Meldungslänge zu empfangen. Boolesche Daten oder Boolesche Felder werden nicht unterstützt. (Standardwert: 0)
NDR	OUT	Bool	Eine Ausführung lang WAHR, wenn neue Daten bereit sind und die Anweisung fehlerfrei beendet wurde.
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die Anweisung mit Fehler beendet wurde.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)
LENGTH	OUT	UInt	Länge der ausgegebenen Meldung in Byte (Standardwert: 0)

Der STATUS-Wert ist gültig, wenn NDR oder ERROR im Zustand WAHR ist. Der STATUS-Wert liefert den Grund für die Beendigung der Empfangsoperation im CM oder CB. Dies ist typischerweise ein positiver Wert, der angibt, dass die Empfangsoperation erfolgreich war und dass der Empfangsvorgang normal beendet wurde. Ist der STATUS-Wert negativ (das höchstwertige Bit des Hexadezimalwerts ist gesetzt), wurde die Empfangsoperation wegen einer Fehlerbedingung wie Paritäts-, Framing- oder Überlauffehler beendet.


Jede PtP-Kommunikationsschnittstelle kann maximal 1024 Bytes puffern. Hierbei kann es sich um eine große oder mehrere kleinere Meldungen handeln. Sind mehrere Meldungen im CM oder CB verfügbar, gibt die Anweisung RCV\_PTP die älteste verfügbare Meldung aus. Eine nachfolgend ausgeführte Anweisung RCV\_PTP gibt die zweitälteste Meldung aus.

Tabelle 12- 25 Bedingungscode

STATUS (W#16#...)	Beschreibung
0000	Kein Puffer vorhanden
80E0	Meldung wurde beendet, weil der Empfangspuffer voll ist
80E1	Meldung wurde wegen Paritätsfehler beendet
80E2	Meldung wurde wegen Framingfehler beendet
80E3	Meldung wurde wegen Überlauffehler beendet
80E4	Meldung wurde beendet, weil die berechnete Länge die Puffergröße überschreitet
80E5	Umgekehrtes Biassignal (Drahtbruch)
0094	Meldung wurde beendet, weil die maximale Zeichenlänge empfangen wurde
0095	Meldung wurde wegen Meldungs-Timeout beendet
0096	Meldung wurde wegen Zeichenabstands-Timeout beendet
0097	Meldung wurde wegen Antwort-Timeout beendet
0098	Meldung wurde beendet, weil die Längenbedingung "N+LEN+M" erfüllt war
0099	Meldung wurde beendet, weil die Endezeichenfolge erfüllt war
833A	Der DB für den Parameter BUFFER ist nicht vorhanden.

### 12.3.1.7 Operation RCV\_RST

Tabelle 12- 26 Anweisung RCV\_RST (Empfänger zurücksetzen)

KOP/FUP	SCL	Beschreibung
	<pre>"RCV_RST_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	RCV_RST löscht den Empfangspuffer im CM oder CB.


<sup>1</sup> STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 12- 27 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Aktiviert das Löschen des Empfangspuffers bei steigender Flanke dieses Freigabeeingangs (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablentabelle zugewiesen. (Standardwert: 0)
DONE	OUT	Bool	Einen Zyklus lang WAHR bedeutet, dass die letzte Anforderung fehlerfrei ausgeführt wurde.
ERROR	OUT	Bool	WAHR bedeutet, dass die letzte Anforderung mit Fehlern ausgeführt wurde. Ist dieser Ausgang WAHR, enthält Ausgang STATUS die zugehörigen Fehlercodes.
STATUS	OUT	Word	Fehlercode (Standardwert: 0)

### 12.3.1.8 Operation SGN\_GET

Tabelle 12- 28 Anweisung SGN\_GET (RS232-Signale abrufen)

KOP/FUP	SCL	Beschreibung
	<pre>"SGN_GET_DB" (     REQ:= _bool_in_,     PORT:= _uint_in_,     NDR=&gt; _bool_out_,     ERROR=&gt; _bool_out_,     STATUS=&gt; _word_out_,     DTR=&gt; _bool_out_,     DSR=&gt; _bool_out_,     RTS=&gt; _bool_out_,     CTS=&gt; _bool_out_,     DCD=&gt; _bool_out_,     RING=&gt; _bool_out_ );</pre>	<p>SGN_GET liest die aktuellen Zustände der RS232-Kommunikationssignale.</p> <p>Diese Funktion gilt nur beim RS232-CM.</p>

1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 12- 29 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	RS232-Signalzustände werden bei der steigenden Flanke an diesem Eingang abgerufen (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablentabelle zugewiesen.
NDR	OUT	Bool	Einen Zyklus lang WAHR, wenn neue Daten bereit sind und die Anweisung fehlerfrei beendet wurde



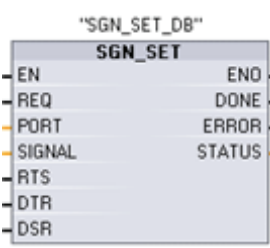
Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die Anweisung mit Fehler beendet wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)
DTR	OUT	Bool	Datenterminal bereit, Modul bereit (Ausgang). Standardwert: False
DSR	OUT	Bool	Datensatz bereit, Kommunikationsteilnehmer bereit (Eingang). Standardwert: False
RTS	OUT	Bool	Sendeanforderung, Modul sendebereit (Ausgang). Standardwert: False
CTS	OUT	Bool	Sendebereit, Kommunikationsteilnehmer kann Daten empfangen (Eingang). Standardwert: False
DCD	OUT	Bool	Datenträger erkannt, Signalpegel empfangen (immer False, nicht unterstützt)
RING	OUT	Bool	Rufanzeige, Meldung eines eingehenden Rufs (immer False, nicht unterstützt)

Tabelle 12- 30 Bedingungscode

STATUS (W#16#....)	Beschreibung
80F0	CM oder CB ist RS485 und es sind keine Signale verfügbar

### 12.3.1.9 Operation SGN\_SET

Tabelle 12- 31 Anweisung SGN\_SET (RS232-Signale setzen)

KOP/FUP	SCL	Beschreibung
	<pre>"SGN_SET_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   SIGNAL:=_byte_in_,   RTS:=_bool_in_,   DTR:=_bool_in_,   DSR:=_bool_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>SGN_SET setzt die Zustände der RS232-Kommunikationssignale.</p> <p>Diese Funktion gilt nur beim RS232-CM.</p>

1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 12- 32 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
REQ	IN	Bool	Die Anweisung zum Setzen der RS232-Signale wird bei der steigenden Flanke an diesem Eingang gestartet (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)

Parameter und Datentyp		Datentyp	Beschreibung
SIGNAL	IN	Byte	Auswahl des zu setzenden Signals: (mehrere zulässig). Standardwert: 0 <ul style="list-style-type: none"> <li>• 01H = RTS</li> <li>• 02H = DTR</li> <li>• 04H = DSR</li> </ul>
RTS	IN	Bool	Sendeanforderung, Modul bereit, zu setzenden Wert zu senden (wahr oder falsch), Standardwert: Falsch
DTR	IN	Bool	Datenterminal bereit, Modul bereit, zu setzenden Wert zu senden (wahr oder falsch). Standardwert: Falsch
DSR	IN	Bool	Datensatz bereit (gilt nur für Schnittstellentyp DCE), nicht verwendet.
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle 12- 33 Bedingungscode

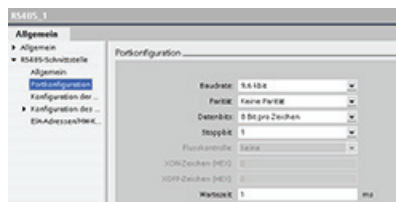
STATUS (W#16#....)	Beschreibung
80F0	CM oder CB ist RS485 und es können keine Signale gesetzt werden
80F1	Signale können wegen Hardwareflusskontrolle nicht gesetzt werden
80F2	DSR kann nicht gesetzt werden, weil das Modul ein DTE-Gerät ist
80F3	DTR kann nicht gesetzt werden, weil das Modul ein DCE-Gerät ist

### 12.3.2 Kommunikationsanschlüsse konfigurieren

Die Kommunikationsschnittstellen können auf zwei Arten konfiguriert werden:

- Konfigurieren Sie in der Gerätekonfiguration in STEP 7 die Portparameter (Baudrate und Parität), die Sendeparameter und die Empfangsparameter. Die Einstellungen der Gerätekonfiguration werden in der CPU gespeichert. Diese Einstellungen werden nach einem Neustart oder nach einem Wechsel von RUN in STOP angewendet.
- Richten Sie die Parameter über die Anweisungen PORT\_CFG (Seite 602), SEND\_CFG (Seite 604) und RCV\_CFG (Seite 605) ein. Die von den Anweisungen festgelegten Porteeinstellungen sind gültig, wenn sich die CPU in der Betriebsart RUN befindet. Die Porteeinstellungen kehren nach einem Wechsel in STOP oder nach einem Neustart zu den Einstellungen der Gerätekonfiguration zurück.

Nach der Konfiguration der Hardwaregeräte (Seite 129) parametrieren Sie die Kommunikationsschnittstellen durch Auswahl eines der CMs in Ihrem Baugruppenträger bzw. des CBs, sofern konfiguriert.



Im Inspektorfenster werden im Register "Eigenschaften" die Parameter des ausgewählten CMs oder CB angezeigt. Wählen Sie "Portkonfiguration", um die folgenden Parameter zu bearbeiten:

- Baudrate
- Parität
- Datenbits pro Zeichen
- Anzahl Stoppsbits
- Flusskontrolle (nur RS232)
- Wartezeit

Beim CM 1241 RS232 und RS485 und beim CB RS485 (außer für die Flusskontrolle, die nur vom CM 1241 RS232 unterstützt wird) haben die RS232- und RS485-Kommunikationsmodule bzw. das RS485-Kommunikationsboard die gleichen Parameter für die Portkonfiguration. Die Parameterwerte können verschieden sein.

Beim CM 1241 RS422/485 wird die Portkonfiguration, wie in der folgenden Abbildung gezeigt, unterschiedlich vorgenommen. Der 422-Modus des CM 1241 RS422/485 unterstützt ebenfalls die Software-Flusskontrolle.



Wählen Sie "Portkonfiguration", um die folgenden RS422/485-Parameter zu bearbeiten:

- "Betriebsart":
  - Vollduplex (RS422), Vierdrahtmodus (Punkt-zu-Punkt-Verbindung)
  - Vollduplex (RS422), Vierdrahtmodus (Multipoint-Master)
  - Vollduplex (RS422), Vierdrahtmodus (Multipoint-Slave)
  - Halbduplex (RS485), Zweidrahtmodus
- "Ausgangszustand Empfangsleitung":
  - Ohne
  - Vorspannung in Vorwärtsrichtung (Signal R(A) 0 V, Signal R(B) 5 V)

Das STEP 7 Anwenderprogramm kann auch den Port konfigurieren oder die vorhandene Konfiguration mit der Anweisung PORT\_CFG (Seite 602) ändern.

### Hinweis

Parameterwerte, die mit PORT\_CFG im Anwenderprogramm eingegeben wurden, überschreiben die Konfigurationseinstellungen der Schnittstelle in der Gerätekonfiguration. Parameter, die mit PORT\_CFG eingegeben wurden, werden in der S7-1200 bei einer Abschaltung nicht gespeichert.

Parameter	Definition
Baudrate	Die Voreinstellung für die Baudrate ist 9,6 kBit/s. Gültige Werte sind: 300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s, 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s und 115,2 kBit/s.
Parität	Die Voreinstellung für die Parität ist "Keine". Gültige Werte sind: Keine Parität, gerade, ungerade, Mark (Paritätsbit immer auf 1) und Space (Paritätsbit immer auf 0).
Datenbits pro Zeichen	Die Anzahl der Datenbits in einem Zeichen. Gültig sind die Einstellungen 7 oder 8.
Anzahl Stoppbits	Die Anzahl der Stoppbits kann eins oder zwei sein. Die Voreinstellung ist eins.
Flusskontrolle	Für das RS232-Kommunikationsmodul kann die Hardware- oder Software-Flusskontrolle eingestellt werden, siehe Beschreibung im Kapitel "Flusskontrolle verwalten (Seite 620)". Ist die Hardware-Flusskontrolle eingestellt, so können Sie wählen, ob das RTS-Signal immer ein ist oder ob RTS geschaltet wird. Ist die Software-Flusskontrolle eingestellt, können Sie die Zeichen XON und XOFF definieren.  Die RS485-Kommunikationsschnittstellen unterstützen keine Flusskontrolle. Der 422-Modus des CM 1241 RS422/485 unterstützt die Software-Flusskontrolle.
Wartezeit	Die Wartezeit ist die Zeit, während der das CM oder CB nach dem Absetzen von RTS auf den Empfang von CTS oder nach dem Empfang von XOFF auf den Empfang von XON, je nach Art der Flusskontrolle, wartet. Ist die Wartezeit beendet, bevor die Kommunikationsschnittstelle ein erwartetes CTS oder XON empfängt, so wird die Sendung durch das CM oder CB abgebrochen und eine Fehlermeldung an das Anwenderprogramm ausgegeben. Geben Sie die Wartezeit in Millisekunden ein. Der zulässige Wertebereich ist 0 bis 65535 Millisekunden.
Betriebszustand	Hier wählen Sie den Betriebszustand RS422 oder RS485 und die Netzwerkkonfigurationen aus.
Ausgangszustand Empfangsleitung	Hier wählen Sie die Vorspannung aus. Gültig sind die Werte Ohne, Vorspannung in Vorwärtsrichtung und Vorspannung in Sperrrichtung. Vorspannung in Sperrrichtung ermöglicht die Drahtbrucherkennung.

### 12.3.2.1 Steuerung der Flusskontrolle

Die Flusskontrolle sorgt für den Ausgleich zwischen unterschiedlichen Kapazitäten von Absender und Empfänger der Daten, so dass keine Daten verlorengehen. Mit der Flusskontrolle ist gewährleistet, dass der Absender nicht mehr Informationen überträgt als der Empfänger verarbeiten kann. Die Flusskontrolle ist auf der Hardware- oder der Softwareebene möglich. Das RS232-Kommunikationsmodul unterstützt die Hardware- und Software-Flusskontrolle. Die RS485-CM und -CB unterstützen keine Flusskontrolle. Der 422-Modus des CM 1241 RS422/485 unterstützt die Software-Flusskontrolle. Welche Flusskontrolle eingerichtet werden soll, geben Sie beim Konfigurieren der Schnittstelle (Seite 618) oder über die Anweisung PORT\_CFG (Seite 602) ein.

Die Hardware-Flusskontrolle beruht auf den Kommunikationssignalen RTS (Request-to-send) und CTS (Clear-to-send). Beim RS232-Kommunikationsmodul wird das RTS-Signal an Pin 7 und das CTS-Signal an Pin 8 abgenommen. Das RS232-Kommunikationsmodul ist ein DTE-Gerät (Data Terminal Equipment, Datenendgerät), das RTS als Ausgang sicherstellt und CTS als Eingang überwacht.

### Hardware-Flusskontrolle: RTS-geschaltet

Ist die RTS-geschaltete Hardware-Flusskontrolle für ein RS232-Kommunikationsmodul aktiviert, setzt das Modul das RTS-Signal auf 1, damit Daten gesendet werden. Es überwacht das CTS-Signal und erkennt damit, ob das Empfangsgerät empfangsbereit ist. Ist das CTS-Signal aktiv, kann das Modul Daten übertragen, solange das CTS-Signal aktiv bleibt. Wird das CTS-Signal inaktiv, muss die Übertragung angehalten werden.

Sie wird fortgesetzt, wenn das CTS-Signal erneut eingeschaltet wird. Wird das CTS-Signal nicht innerhalb der eingestellten Wartezeit erneut eingeschaltet, bricht das Modul die Übertragung ab und gibt einen Fehler an das Anwenderprogramm zurück. Die Wartezeit müssen Sie in der Schnittstellenkonfiguration (Seite 618) eingeben.

Die RTS-geschaltete Flusskontrolle ist bei Geräten nützlich, die ein Signal benötigen, dass der Sendevorgang aktiv ist. Ein Beispiel hierfür ist ein Funkmodem, bei dem RTS als "Schlüssel"signal den Funksender aktiviert. Die RTS-geschaltete Flusskontrolle funktioniert nicht bei herkömmlichen Telefonmodems. Verwenden Sie für Telefonmodems die Option "RTS immer ein".

### Hardware-Flusskontrolle: RTS immer ein

Bei der Option "RTS immer ein" aktiviert das CM 1241 das RTS-Signal standardmäßig. Ein Gerät, wie z. B. ein Telefonmodem, überwacht das RTS-Signal vom CM und nutzt dieses Signal als Mitteilung der Sendebereitschaft. Das Modem sendet nur dann an das CM, wenn RTS aktiv ist, d. h. wenn das Telefonmodem ein aktives CTS erkennt. Ist RTS inaktiv, überträgt das Telefonmodem keine Daten zum CM.

Damit das Modem jederzeit Daten an das CM senden kann, ist für die Hardware-Flusskontrolle "RTS immer ein" einzustellen. Das CM setzt das RTS-Signal dann immer auf 1. Das CM schaltet RTS auch dann nicht aus, wenn das Modul keine Zeichen empfangen kann. Das Sendegerät muss gewährleisten, dass es keinen Überlauf im Empfangspuffer des CMs verursacht.

### Signale Data Terminal Block Ready (DTR) und Data Set Ready (DSR)

Das CM setzt DTR für jede Art der Hardware-Flusskontrolle auf 1. Es überträgt erst, wenn das Signal DSR eingeschaltet wird. Der Zustand von DSR wird nur am Anfang der Sendefunktion ausgewertet. Wenn DSR nach Beginn der Übertragung inaktiv wird, wird die Übertragung nicht angehalten.

### Software-Flusskontrolle

Bei der Software-Flusskontrolle werden spezielle Zeichen in den Meldungen für die Flusskontrolle verwendet. Sie können Hexadezimalzeichen konfigurieren, die XON und XOFF darstellen.

XOFF zeigt an, dass eine Übertragung angehalten werden muss. XON zeigt an, dass eine Übertragung fortgesetzt werden kann. XOFF und XON dürfen nicht das gleiche Zeichen sein.

Empfängt das Sendegerät ein Zeichen XOFF vom Empfänger, wird die Übertragung angehalten. Sie wird fortgesetzt, wenn das Sendegerät das Zeichen XON empfängt. Empfängt das CM innerhalb der in der Schnittstellenkonfiguration (Seite 618) angegebenen Wartezeit kein XON-Zeichen, bricht es die Übertragung ab und gibt eine Fehlermeldung an das Anwenderprogramm aus.

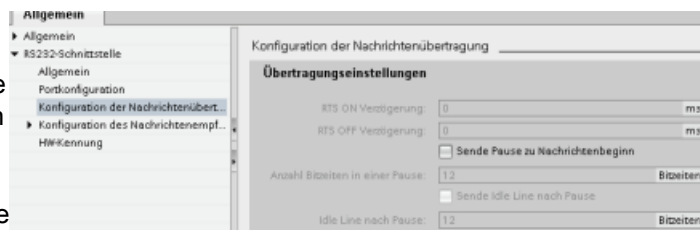
Die Software-Flusskontrolle setzt die Vollduplex-Kommunikation voraus, da der Empfänger in der Lage sein muss, während einer laufenden Übertragung XOFF an den Sender zu senden. Die Software-Flusskontrolle ist nur bei Meldungen möglich, die nur aus ASCII-Zeichen bestehen. Binäre Protokolle können keine Software-Flusskontrolle nutzen.

### 12.3.3 Sende- und Empfangsparameter konfigurieren

Damit die Punkt-zu-Punkt-Kommunikation aufgenommen werden kann, müssen die Parameter zum Senden und Empfangen von Meldungen eingerichtet werden. Diese Parameter legen fest, wie die Kommunikation abläuft, wenn Meldungen von einem Zielgerät gesendet oder von diesem empfangen werden.

#### 12.3.3.1 Sendeparameter konfigurieren

In der Gerätekonfiguration richten Sie ein, wie eine Kommunikationsschnittstelle Daten sendet. Hierfür geben Sie die Eigenschaften "Konfiguration der gesendeten Meldung" für die ausgewählte Schnittstelle an.



Sie können ferner die Parameter für das Senden einer Meldung im Anwenderprogramm mit der Anweisung SEND\_CFG (Seite 604) dynamisch konfigurieren oder ändern.

---

#### Hinweis

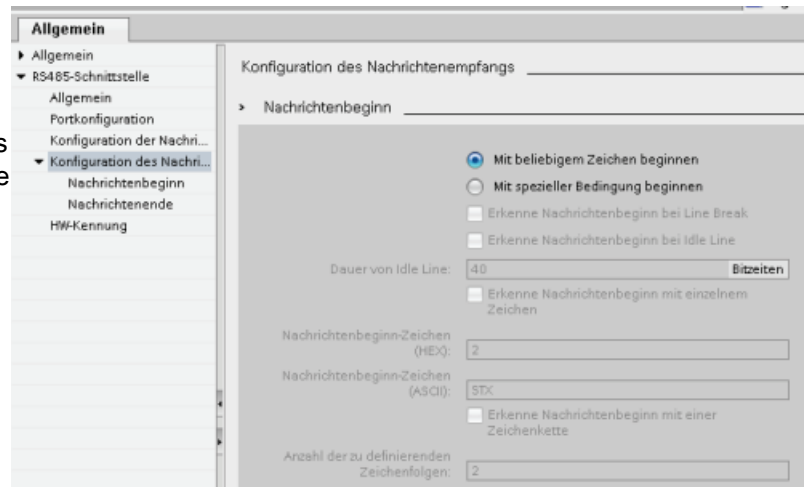
Parameterwerte, die mit SEND\_CFG im Anwenderprogramm eingegeben wurden, überschreiben die Konfigurationseinstellungen der Schnittstelle. Parameter, die mit der Anweisung SEND\_CFG eingegeben wurden, werden bei einer Abschaltung nicht in der CPU gespeichert.

---

Parameter	Definition
RTS-Einschaltverzögerung	Wartezeit nach dem Aktivieren von RTS, bis die Sendung ausgelöst wird. Der gültige Bereich ist 0 bis 65535 ms, voreingestellt ist 0. Dieser Parameter ist nur gültig, wenn in der Schnittstellenkonfiguration (Seite 618) die Hardware-Flusskontrolle eingestellt ist. CTS wird nach dem Ablauf der RTS-Einschaltverzögerung bearbeitet. Dieser Parameter gilt nur für RS232-Module.
RTS-Ausschaltverzögerung	Wartezeit nach der beendeten Übertragung, bis RTS deaktiviert wird. Der gültige Bereich ist 0 bis 65535 ms, voreingestellt ist 0. Dieser Parameter ist nur gültig, wenn in der Schnittstellenkonfiguration (Seite 618) die Hardware-Flusskontrolle eingestellt ist. Dieser Parameter gilt nur für RS232-Module.
Sendepause bei Meldungsbeginn Anzahl der Bittakte in einer Pause	Damit wird festgelegt, dass beim Start einer Meldung eine Pause gesendet wird, wenn die RTS -inschaltverzögerung (sofern eingerichtet) abgelaufen ist und CTS aktiv ist. Wie viele Bittakte eine Pause darstellen, während der die Übertragung angehalten ist, können Sie einstellen. Die Voreinstellung ist 12 und das Maximum ist 65535 bis zu einem Grenzwert von acht Sekunden.
Leerzeile nach Pause senden Leerzeile nach Pause	Gibt an, dass eine Leerzeile vor Meldungsbeginn gesendet wird. Sie wird nach der Pause gesendet, sofern eine Pause konfiguriert ist. Der Parameter "Leerzeile nach Pause" gibt an, wie viele Bittakte einer Leerzeile entsprechen. Die Voreinstellung ist 12 und das Maximum ist 65535 bis zu einem Grenzwert von acht Sekunden.

### 12.3.3.2 Empfangsparameter konfigurieren

In der Gerätekonfiguration legen Sie fest, wie Daten über eine Kommunikationsschnittstelle empfangen und wie Meldungsbeginn und Meldungsende erkannt werden sollen. Geben Sie diese Parameter in der Konfiguration der Empfangsmeldung für die ausgewählte Schnittstelle an.



Sie können ferner die Meldungsparameter im Anwenderprogramm mit der Anweisung RCV\_CFG (Seite 605) dynamisch konfigurieren oder ändern.

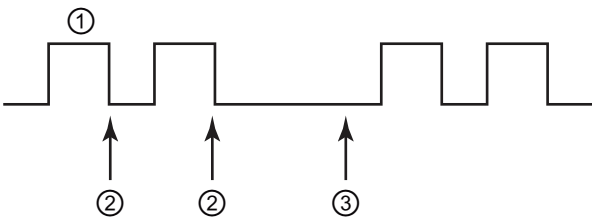
#### Hinweis

Parameterwerte, die mit der Anweisung RCV\_CFG im Anwenderprogramm eingegeben wurden, überschreiben die Konfigurationseinstellungen der Schnittstelle. Parameter, die mit der Anweisung RCV\_CFG eingegeben wurden, werden bei einer Abschaltung nicht in der CPU gespeichert.

### Bedingungen für den Meldungsbeginn

Sie können festlegen, wie die Kommunikationsschnittstelle den Meldungsbeginn erkennen soll. Startzeichen und Meldungsinhalt werden im Empfangspuffer abgelegt, bis eine konfigurierte Endebedingung erfüllt ist.

Sie können mehrere Startbedingungen angeben. Wenn Sie mehrere Startbedingungen angeben, müssen alle Startbedingungen erfüllt sein, damit die Meldung als gestartet betrachtet wird. Wenn Sie z. B. eine Zeit für den Leitungsleerlauf und ein bestimmtes Startzeichen konfigurieren, wartet das CM bzw. CB zunächst auf die Erfüllung der Zeit für den Leitungsleerlauf und sucht dann nach dem spezifischen Startzeichen. Wenn ein anderes Zeichen empfangen wird (nicht das angegebene Startzeichen), startet das CM bzw. CB die Suche nach dem Meldungsstart erneut, indem es auf die Zeit für den Leitungsleerlauf wartet.

Parameter	Definition
Start mit beliebigem Zeichen	Die Bedingung "Beliebiges Zeichen" gibt an, dass durch erfolgreichen Empfang eines beliebigen Zeichens ein Meldungsbeginn angezeigt wird. Dieses Zeichen ist das erste Zeichen in einer Meldung.
Leitungspause	Die Bedingung "Leitungspause" gibt an, dass eine Operation zum Meldungsempfang beginnt, nachdem ein Pausezeichen empfangen wurde.
Leitung im Leerlauf	<p>Die Bedingung "Leitung im Leerlauf" gibt an, dass der Meldungsempfang beginnt, wenn die Empfangsleitung für die Anzahl der angegebenen Bitzeiten im Leerlauf war. Nachdem diese Bedingung aufgetreten ist, beginnt der Empfang einer Meldung.</p>  <p>① Zeichen                  ② Startet die Zeit für den Leitungsleerlauf neu                  ③ Leitungsleerlauf wird erkannt und Meldungsempfang gestartet</p>
Sonderbedingung: Meldungsbeginn durch einzelnes Zeichen erkennen	Gibt an, dass ein bestimmtes Zeichen den Beginn einer Meldung anzeigt. Dieses Zeichen ist dann das erste Zeichen in einer Meldung. Jedes Zeichen, das vor diesem bestimmten Zeichen empfangen wird, wird ignoriert. Das Standardzeichen ist STX.
Sonderbedingung: Meldungsbeginn durch Zeichenfolge erkennen	<p>Gibt an, dass eine bestimmte Zeichenfolge aus bis zu vier konfigurierten Folgen den Beginn einer Meldung anzeigt. Sie können bis zu fünf Zeichen für jede Zeichenfolge eingeben. Für jede Zeichenposition kann ein bestimmtes Hex-Zeichen oder das Ignorieren des Zeichens in der Zeichenfolge (Platzhalterzeichen) eingegeben werden. Das letzte spezifische Zeichen einer Zeichenfolge beendet die Startbedingungsfolge.</p> <p>Eingehende Zeichenfolgen werden mit den konfigurierten Startbedingungen verglichen, bis eine Startbedingung erfüllt ist. Ist die Startbedingung erfüllt, beginnt die Überwachung der Endebedingung.</p> <p>Sie können bis zu vier spezifische Zeichenfolgen konfigurieren. Sie verwenden eine Startbedingung mit mehreren Folgen, wenn verschiedene Zeichenfolgen den Beginn einer Meldung anzeigen können. Die Meldung beginnt, wenn eine der eingerichteten Zeichenfolgen erfüllt ist.</p>



Die Startbedingungen werden in der folgenden Reihenfolge geprüft:

- Leerzeile
- Zeilenumbruch
- Zeichen oder Zeichenfolgen

Wenn mehrere Startbedingungen geprüft werden und eine der Bedingungen nicht erfüllt wird, beginnt das CM bzw. CB die Prüfung erneut mit der ersten geforderten Bedingung. Nachdem das CM bzw. CB ermittelt hat, dass die Startbedingungen erfüllt wurden, beginnt es mit dem Auswerten der Endbedingungen.

### Beispielkonfiguration - Meldungsbeginn bei einer von zwei Zeichenfolgen

Beachten Sie die folgende Konfiguration der Bedingungen für den Meldungsbeginn:

Erkenne Nachrichtenbeginn mit einer Zeichenkette

Anzahl der zu definierenden Zeichenfolgen: 2

#### 5-Zeichen-Sequenz zum Nachrichtenbeginn

##### Nachrichtenbeginn Sequenz 1

<input checked="" type="checkbox"/> Prüfe Zeichen 1
Zeichenwert (HEX): 6A
Zeichenwert (ASCII): j
<input type="checkbox"/> Prüfe Zeichen 2
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Prüfe Zeichen 3
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Prüfe Zeichen 4
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input checked="" type="checkbox"/> Prüfe Zeichen 5
Zeichenwert (HEX): 1C
Zeichenwert (ASCII): FS

Nachrichtenbeginn Sequenz 2

Prüfe Zeichen 1  
Zeichenwert (HEX): 0  
Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 2  
Zeichenwert (HEX): 6A  
Zeichenwert (ASCII):

Prüfe Zeichen 3  
Zeichenwert (HEX): 6A  
Zeichenwert (ASCII):

Prüfe Zeichen 4  
Zeichenwert (HEX): 0  
Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 5  
Zeichenwert (HEX): 0  
Zeichenwert (ASCII): BELIEBIG

Bei dieser Konfiguration ist die Startbedingung erfüllt, wenn eines der folgenden Bitmuster vorliegt:

- Empfang einer Zeichenfolge mit 5 Zeichen, bei der das erste Zeichen 0x6A und das fünfte Zeichen 0x1C ist. Die Zeichen 2, 3 und 4 können hierbei beliebig sein. Nach dem Empfang des fünften Zeichens beginnt die Überwachung auf die Endbedingung.
- Empfang von zwei aufeinanderfolgenden Zeichen 0x6A nach einem beliebigen anderen Zeichen. In diesem Fall beginnt die Überwachung auf die Endbedingung nach dem Empfang der zweiten 0x6A-Folge (3 Zeichen). Das Zeichen vor dem ersten 0x6A ist in der Startbedingung enthalten.

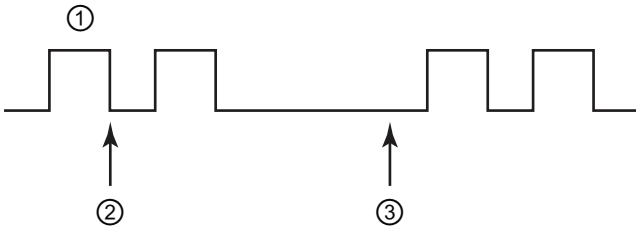
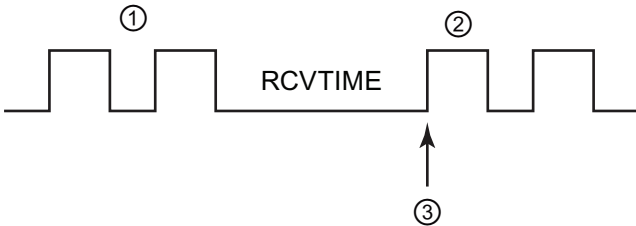
Beispiele für Zeichenfolgen, die diese Startbedingung erfüllen:

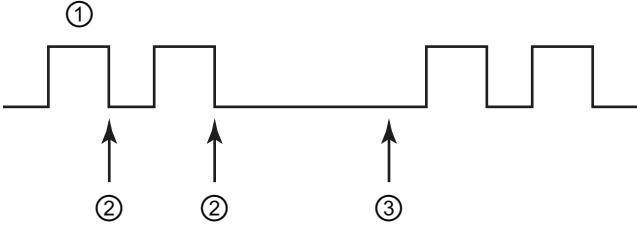
- <beliebiges Zeichen> 6A 6A
- 6A 12 14 18 1C
- 6A 44 A5 D2 1C

### Bedingungen für das Meldungsende

Sie legen ferner fest, wie die Kommunikationsschnittstelle das Meldungsende erkennen soll. Sie können mehrere Endbedingungen für Meldungen einrichten. Die Meldung endet, sobald eine der eingerichteten Bedingungen erfüllt ist.

Sie können z. B. eine Endbedingung mit einem Meldungs-Timeout von 300 ms, einem Zeichenabstands-Timeout von 40 Bitzeiten und eine maximale Länge von 50 Bytes angeben. Die Meldung endet, wenn der Meldungsempfang länger als 300 ms dauert oder wenn der Abstand zwischen zwei beliebigen Zeichen 40 Bitzeiten überschreitet oder wenn 50 Bytes empfangen wurden.

Parameter	Definition
Meldungsende erkennen durch Meldungs-Timeout	<p>Das Meldungsende ist erreicht, wenn die konfigurierte Wartezeit auf das Meldungsende abgelaufen ist. Die Timeout-Zeit der Meldung beginnt, wenn eine Startbedingung erfüllt wurde. Die Voreinstellung ist 200 ms, der Wertebereich ist 0 bis 65535 ms.</p>  <p>① Empfangene Zeichen          ② Bedingung für den Meldungsbeginn erfüllt: Meldungs-Timer beginnt zu laufen          ③ Meldungs-Timer läuft ab und beendet die Meldung</p>
Meldungsende erkennen durch Antwort-Timeout	<p>Das Meldungsende ist erreicht, wenn die konfigurierte Wartezeit auf eine Antwort erreicht ist, bevor eine gültige Startzeichenfolge empfangen wurde. Die Timeout-Zeit für die Antwort beginnt, wenn eine Übertragung endet und das CM bzw. CB die Empfangsanweisung startet. Die Voreinstellung für die Timeout-Zeit der Antwort beträgt 200 ms, der Wertebereich ist 0 bis 65535 ms. Wenn nicht innerhalb des von RCVTIME angegebenen Zeitraums für die Antwort ein Zeichen empfangen wird, wird der entsprechenden Anweisung RCV_PTP ein Fehler gemeldet. Das Antwort-Timeout definiert keine spezifische Endebedingung. Das Timeout gibt nur an, dass ein Zeichen innerhalb der angegebenen Zeit erfolgreich empfangen werden muss. Sie müssen eine andere Endebedingung konfigurieren, um das tatsächliche Ende einer Meldung anzugeben.</p>  <p>① Gesendete Zeichen          ② Empfangene Zeichen          ③ Erstes Zeichen muss zu diesem Zeitpunkt erfolgreich empfangen worden sein.</p>

Parameter	Definition
Meldungsende durch Zeichenabstand erkennen	<p>Das Meldungsende ist erreicht, wenn die konfigurierte maximale Zeit zwischen zwei beliebigen, aufeinanderfolgenden Zeichen einer Meldung abgelaufen ist. Die Voreinstellung für den Zeichenabstand ist 12 Bittakte und die maximale Zahl ist 65535 Bittakte, bis maximal acht Sekunden.</p>  <p>① Empfangene Zeichen                  ② Startet den Timer für die Zeit zwischen den Zeichen neu                  ③ Der Timer für die Zeit zwischen den Zeichen läuft ab und beendet die Meldung.</p>
Meldungsende durch maximale Meldungslänge erkennen	<p>Das Meldungsende ist erreicht, wenn die konfigurierte maximale Zeichenzahl empfangen wurde. Der gültige Bereich für die maximale Länge liegt zwischen 1 und 1023.</p> <p>Diese Bedingung kann dafür genutzt werden, den Fehler "Überlauf Meldungspuffer" zu verhindern. Wird diese Endebedingung mit Timeout-Endebedingungen verknüpft und tritt die Timeout-Endebedingung auf, werden alle gültigen empfangenen Zeichen ausgegeben, auch wenn die maximale Länge nicht erreicht wurde. So können Protokolle unterschiedlicher Länge unterstützt werden, auch wenn nur die maximale Länge bekannt ist.</p>
Meldungslänge aus Meldung lesen	<p>Die Meldungslänge ist in der Meldung selbst angegeben. Das Meldungsende ist erreicht, wenn eine Meldung mit der angegebenen Länge empfangen wurde. Wie die Meldungslänge angegeben und ausgewertet wird, ist nachstehend beschrieben.</p>
Meldungsende mit einem Zeichen erkennen	<p>Das Meldungsende ist erreicht, wenn ein bestimmtes Zeichen empfangen wird.</p>
Meldungsende mit einer Zeichenfolge erkennen	<p>Das Meldungsende ist erreicht, wenn eine bestimmte Zeichenfolge empfangen wird. Sie können eine Zeichenfolge mit bis zu 5 Zeichen angeben. Für jede Zeichenposition kann ein bestimmtes Hex-Zeichen oder dass das Zeichen in der Zeichenfolge ignoriert werden soll eingegeben werden.</p> <p>Führende Zeichen, die ignoriert werden, gehören nicht zur Endebedingung. Nachgestellte Zeichen, die ignoriert werden, gehören zur Endebedingung.</p>

## Beispielkonfiguration - Meldung mit einer Zeichenfolge beenden

Beachten Sie die folgende Konfiguration der Bedingungen für das Meldungsende:

Die Endebedingung ist in diesem Fall erfüllt, wenn zwei aufeinanderfolgende Zeichen 0x7A, gefolgt von zwei beliebigen Zeichen, empfangen werden. Das Zeichen vor 0x7A 0x7A gehört nicht zur Endezeichenfolge. Nach 0x7A 0x7A sind zwei Zeichen erforderlich, um die Endezeichenfolge abzuschließen. Die von Zeichen 4 und 5 empfangenen Werte sind irrelevant, sie müssen jedoch empfangen werden, damit die Endebedingung erfüllt ist.

## Angabe der Meldungslänge in der Meldung

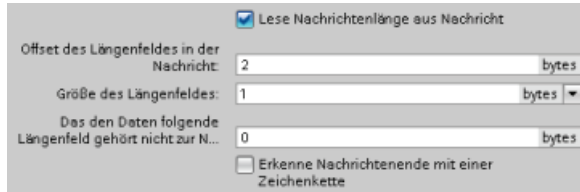
Bei der Auswahl der Sonderbedingung, bei der die Meldungslänge in der Meldung enthalten ist, müssen drei Parameter angegeben werden, die Informationen über die Meldungslänge enthalten.

Die tatsächliche Meldungsstruktur hängt vom verwendeten Protokoll ab. Die drei Parameter sind:

- n: Zeichenposition (Basis 1) innerhalb der Meldung, an der die Längenangabe beginnt
- Längenwert: Anzahl der Bytes (eins, zwei oder vier) der Längenangabe
- Länge m: Anzahl der Zeichen nach der Längenangabe, die nicht in der Längenzählung berücksichtigt werden

Die Endezeichen müssen nicht aufeinander folgen. Der Wert "Länge m" kann genutzt werden, um die Länge eines Prüfsummenfelds anzugeben, dessen Größe nicht im Längenfeld enthalten ist.

Diese Felder erscheinen in der Konfiguration des Meldungsempfangs in den Geräteeigenschaften:



**Beispiel 1:** Es wird eine Meldung betrachtet, die nach dem folgenden Protokoll strukturiert ist:

STX	Len (n)	Zeichen 3 bis 14, nach Länge gezählt											
		ADR	PKE		INDEX		PWD		STW		HSW		BCC
1	2	3	4	5	6	7	8	9	10	11	12	13	14
STX	0x0C	xx	xxxx		xxxx		xxxx		xxxx		xxxx		xx

Die Parameter für die Länge des Meldungsempfangs konfigurieren Sie wie folgt:

- n = 2 (Meldungslänge startet in Byte 2.)
- Längenswert = 1 (Meldungslänge ist in einem Byte enthalten.)
- Länge m = 0 (Auf die Längenangabe folgen keine weiteren Zeichen, die nicht in der Längenzählung berücksichtigt werden. Nach der Längenangabe folgen zwölf Zeichen.)

In diesem Beispiel werden die Zeichen von 3 bis einschließlich 14 von Len (n) gezählt.

**Beispiel 2:** Es wird eine weitere Meldung betrachtet, die nach dem folgenden Protokoll strukturiert ist:

SD1	Len (n)	Len (n)	SD2	Zeichen 5 bis 10, nach Länge gezählt						FCS	ED
				DA	SA	FA	Dateneinheit=3 Bytes				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Die Parameter für die Länge des Meldungsempfangs konfigurieren Sie wie folgt:

- n = 3 (Meldungslänge startet in Byte 3.)
- Längenswert = 1 (Meldungslänge ist in einem Byte enthalten.)
- Länge m = 3 (Auf die Längenangabe folgen drei Zeichen, die nicht in der Längenzählung berücksichtigt werden. Im Protokoll dieses Beispiels werden Zeichen SD2, FCS und ED bei der Längenzählung nicht beachtet. Die sechs anderen Zeichen werden in der Längenzählung berücksichtigt; die Gesamtzahl der Zeichen nach der Längenangabe ist somit neun.)

In diesem Beispiel werden die Zeichen von 5 bis einschließlich 10 von Len (n) gezählt.

### 12.3.4 Programmieren der PtP-Kommunikation

STEP 7 bietet erweiterte Anweisungen, mit denen das Anwenderprogramm die Punkt-zu-Punkt-Kommunikation über ein im Anwenderprogramm vorgegebenes Protokoll durchführen kann. Diese Anweisungen lassen sich in zwei Kategorien unterteilen:

- Konfigurationsanweisungen
- Kommunikationsanweisungen

#### Konfigurationsanweisungen

Bevor das Anwenderprogramm die PtP-Kommunikation starten kann, müssen die Kommunikationsschnittstelle und die Parameter zum Senden und Empfangen der Daten konfiguriert werden.

Die Schnittstellenkonfiguration und die Meldungskonfiguration können für jedes CM oder CB in der Gerätekonfiguration oder mit den folgenden Anweisungen Ihres Anwenderprogramms durchgeführt werden:

- PORT\_CFG (Seite 602)
- SEND\_CFG (Seite 604)
- RCV\_CFG (Seite 605)

#### Kommunikationsanweisungen

Mit den Anweisungen für die Punkt-zu-Punkt-Kommunikation kann das Anwenderprogramm Meldungen an die Kommunikationsschnittstellen senden und von diesen Meldungen empfangen. Beachten Sie für weitere Informationen zum Übertragen von Daten mit diesen Anweisungen den Abschnitt zur Datenkonsistenz (Seite 163).

Alle PtP-Anweisungen funktionieren asynchron. Mit Hilfe einer Abfragearchitektur kann das Anwenderprogramm den Sende- und Empfangsstatus feststellen. SEND\_PTP und RCV\_PTP können gleichzeitig ausgeführt werden. Die Kommunikationsmodule und das Kommunikationsboard puffern die Sende- und Empfangsmeldungen je nach Bedarf bis zu einer maximalen Puffergröße von 1024 Bytes.

Die CMs und das CB senden und empfangen Meldungen an die bzw. von den Kommunikationsteilnehmern. Das Meldungsprotokoll befindet sich in einem Puffer, der von einer bestimmten Kommunikationsschnittstelle empfangen oder an diese gesendet wird. Puffer und Port sind Parameter der Sende- und Empfangsanweisungen:

- SEND\_PTP (Seite 610)
- RCV\_PTP (Seite 614)

Mit zusätzlichen Anweisungen kann der Empfangspuffer zurückgesetzt und es können spezielle RS232-Signale abgefragt und gesetzt werden:

- RCV\_RST (Seite 615)
- SGN\_GET (Seite 616)
- SGN\_SET (Seite 617)

### 12.3.4.1 Abfragearchitektur

Die S7-1200 Punkt-zu-Punkt-Anweisungen müssen zyklisch/regelmäßig aufgerufen werden, um nach empfangenen Meldungen abzufragen. Die Abfrage des Sendevorgangs meldet dem Anwenderprogramm, wenn die Übertragung beendet ist.

#### Abfragearchitektur: Master

Die typische Sequenz für einen Master ist wie folgt:

1. Die Anweisung SEND\_PTP veranlasst eine Übertragung zum CM oder CB.
2. Die Anweisung SEND\_PTP wird in aufeinander folgenden Zyklen ausgeführt, um den Status des Übertragungsvorgangs abzufragen.
3. Wenn die Anweisung SEND\_PTP meldet, dass die Übertragung beendet ist, kann der Anwendercode den Empfang der Antwort vorbereiten.
4. Die Anweisung RCV\_PTP wird wiederholt ausgeführt, um auf eine Antwort abzufragen. Wenn das CM oder CB eine Antwortmeldung erfasst hat, kopiert die Anweisung RCV\_PTP die Antwort in die CPU und meldet, dass neue Daten empfangen wurden.
5. Das Anwenderprogramm kann die Antwort verarbeiten.
6. Zurück zu Schritt 1 und Wiederholung des Zyklus.

#### Abfragearchitektur: Slave

Die typische Sequenz für einen Slave ist wie folgt:

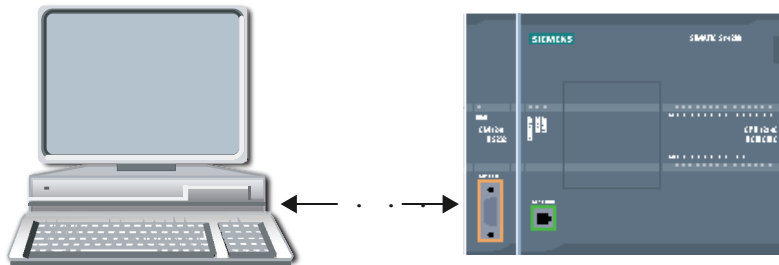
1. Das Anwenderprogramm führt die Anweisung RCV\_PTP in jedem Zyklus aus.
2. Wenn das CM oder CB eine Anforderung empfangen hat, meldet die Anweisung RCV\_PTP, dass neue Daten bereit sind, und die Anforderung wird in die CPU kopiert.
3. Das Anwenderprogramm verarbeitet die Anforderung und erzeugt eine Antwort.
4. Mit der Anweisung SEND\_PTP wird die Antwort an den Master zurückgesendet.
5. Führen Sie SEND\_PTP wiederholt aus, um sicherzustellen, dass der Sendevorgang stattfindet.
6. Zurück zu Schritt 1 und Wiederholung des Zyklus.

Der Slave muss dafür sorgen, dass RCV\_PTP entsprechend oft aufgerufen wird, damit eine Übertragung vom Master empfangen werden kann, bevor dieser beim Warten auf eine Antwort wegen Zeitüberschreitung den Vorgang abbricht. Um diese Aufgabe zu erfüllen, kann das Anwenderprogramm RCV\_PTP aus einem Zyklus-OB heraus aufrufen, dessen Zykluszeit ausreichend lang ist, um eine Übertragung vom Master vor dem Ablauf der Timeout-Einstellung zu empfangen. Wird die Zykluszeit für den OB so eingestellt, dass zwei Ausführungen innerhalb der Timeout-Einstellung des Masters erfolgen, kann das Anwenderprogramm alle Übertragungen ohne Verlust empfangen.



### 12.3.5 Beispiel: Punkt-zu-Punkt-Kommunikation

In diesem Beispiel kommuniziert eine S7-1200 CPU mit einem PC mit einem Terminalemulator über ein CM 1241 RS232-Modul. Die Punkt-zu-Punkt-Konfiguration und das STEP 7-Programm in diesem Beispiel zeigen, wie die CPU eine Meldung vom PC empfangen und das Echo der Meldung an den PC zurückgeben kann.



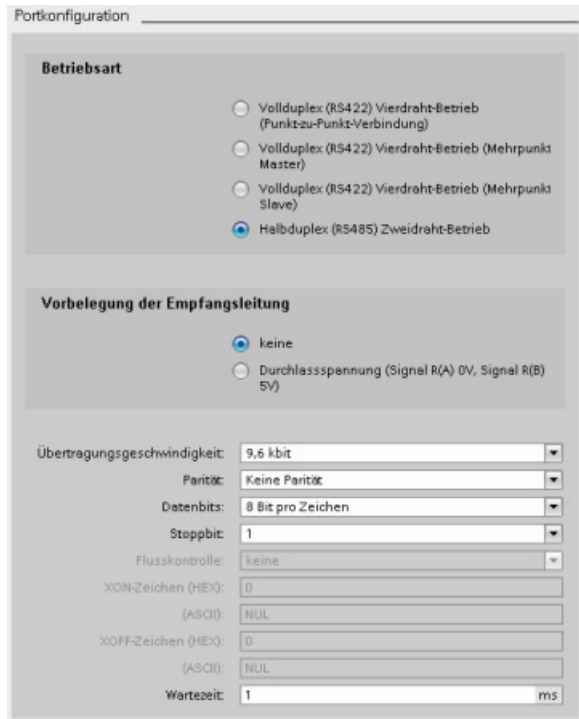
Sie müssen die Kommunikationsschnittstelle des CM 1241 RS232-Moduls an die RS232-Schnittstelle des PC anschließen. Dies ist üblicherweise COM1. Weil es sich bei beiden Ports um Datenendgeräte (Data Terminal Equipment, DTE) handelt, müssen Sie die Empfangs- und Sendepins (2 und 3) schalten, wenn Sie die beiden Ports anschließen. Hierzu gehen Sie auf eine der folgenden Arten vor:

- Verwenden Sie einen NULL-Modemadapter, um die Pins 2 und 3 zu tauschen, zusammen mit einem herkömmlichen RS232-Kabel.
- Verwenden Sie ein NULL-Modemkabel, bei dem die Pins 2 und 3 bereits getauscht sind. Sie erkennen ein NULL-Modemkabel üblicherweise an den zwei 9-poligen D-Buchsen.

### 12.3.5.1 Kommunikationsmodul konfigurieren

Sie können das CM 1241 in der Gerätekonfiguration in STEP 7 oder mit Anweisungen im Anwenderprogramm konfigurieren. In diesem Beispiel wird die Gerätekonfiguration verwendet.

- Schnittstellenkonfiguration: Klicken Sie auf den Kommunikationsport des CM in der Gerätekonfiguration und konfigurieren Sie den Port wie folgt:



#### Hinweis

Die Konfigurationseinstellungen für "Betriebsart" und "Ausgangszustand Empfangsleitung" gelten nur für das Modul CM 1241 (RS422/RS485). Die anderen CM 1241-Module haben diese Portkonfigurationseinstellungen nicht. Weitere Informationen finden Sie unter RS422 und RS485 konfigurieren (Seite 637).

- Konfiguration der Sendemeldung: Übernehmen Sie die Voreinstellung für die Konfiguration zum Senden von Meldungen. Bei Meldungsbeginn wird keine Pause gesendet.

- Beginn des Meldungsempfangs konfigurieren: Konfigurieren Sie das CM 1241 so, dass der Meldungsempfang beginnt, wenn die Kommunikationsleitung mindestens 50 Bitzeiten (ca. 5 ms bei 9600 Baud =  $50 * 1/9600$ ) inaktiv ist:

Nachrichtenbeginn

Mit beliebigem Zeichen beginnen  
 Mit spezieller Bedingung beginnen  
 Erkenne Nachrichtenbeginn bei Line Break  
 Erkenne Nachrichtenbeginn bei Idle Line

Dauer von Idle Line:  Bitzeiten

Erkenne Nachrichtenbeginn mit individuellem Zeichen  
 Erkenne Nachrichtenbeginn mit einer Zeichenkette

Nachrichtenbeginn-Zeichen (HEX):

Nachrichtenbeginn-Zeichen (ASCII):

Anzahl der zu definierenden Zeichenfolgen:

- Ende des Meldungsempfangs konfigurieren: Konfigurieren Sie das CM 1241 so, dass der Meldungsempfang beendet wird, wenn maximal 100 Byte oder ein Zeilenschaltungszeichen (10 dezimal oder A hexadezimal) empfangen wurde. Als Endezeichenfolge sind maximal fünf Endezeichen in Folge zulässig. An der fünften Stelle der Folge befindet sich das Zeilenschaltungszeichen. Die vorhergehenden vier Endezeichen der Folge sind "nicht relevant" oder nicht ausgewählte Zeichen. Das CM 1241 wertet die Zeichen "nicht relevant" nicht aus, erwartet jedoch ein Zeilenschaltungszeichen mit vorhergehender 0 oder weiteren Zeichen "nicht relevant" zur Kennzeichnung des Meldungsendes.

Nachrichtenende

Bedingungen zum Nachrichtenende definieren

Erkenne Nachrichtenende durch Nachrichten-Zeitüberschreitung  
 Erkenne Nachrichtenende durch Antwort-Zeitüberschreitung  
 Erkenne Nachrichtenende durch Zeitüberschreitung innerhalb der Zeichen

Nachrichten-Zeitüberschreitung:  ms

Antwort-Zeitüberschreitung:  ms

Zeichenlücken-Zeitüberschreitung:  Bitzeiten

Erkenne Nachrichtenende durch maximale Länge  
 Lese Nachrichtenlänge aus Nachricht

Maximale Länge der Nachricht:  bytes

Offset des Längenfeldes in der Nachricht:  bytes

Größe des Längenfeldes:  bytes

Das den Daten folgende Längenfeld gehört nicht zur N...:  bytes

Erkenne Nachrichtenende mit einer Zeichenkette

**5-Zeichen-Sequenz zum Nachrichtenende**

<input type="checkbox"/> Frühe Zeichen 1
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Frühe Zeichen 2
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Frühe Zeichen 3
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Frühe Zeichen 4
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input checked="" type="checkbox"/> Frühe Zeichen 5
Zeichenwert (HEX): A
Zeichenwert (ASCII): LF

## 12.3.5.2 RS422 und RS485 konfigurieren

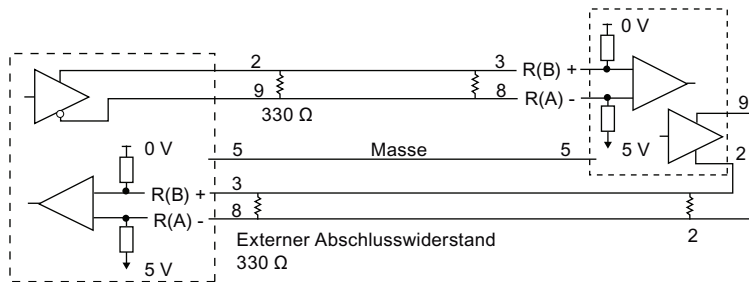
### RS422 konfigurieren

Im RS422-Modus gibt es, abhängig von Ihrer Netzwerkkonfiguration, drei Betriebsarten. Wählen Sie je nach den Geräten in Ihrem Netzwerk eine dieser Betriebsarten aus. Die verschiedenen Einstellungen für "Ausgangszustand Empfangsleitung" beziehen sich auf die im Folgenden dargestellten Fälle.

- Vollduplex (RS422), Vierdrahtmodus (Punkt-zu-Punkt-Verbindung): Wählen Sie diese Option aus, wenn Ihr Netzwerk zwei Geräte umfasst. Für "Ausgangszustand Empfangsleitung":
  - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 3).
  - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 2).
  - Wählen Sie "Vorspannung in Sperrichtung", um interne Vorspannung und Abschluss zu verwenden und die Kabelbrucherkennung für beide Geräte zu aktivieren (Fall 1).
- Vollduplex (RS422), Vierdrahtmodus (Multipoint-Master): Wählen Sie diese Option für das Mastergerät aus, wenn Ihr Netzwerk einen Master und mehrere Slaves umfasst. Für "Ausgangszustand Empfangsleitung":
  - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 3).
  - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 2).
  - Kabelbrucherkennung ist in dieser Betriebsart nicht möglich.
- Vollduplex (RS422), Vierdrahtmodus (Multipoint-Slave): Wählen Sie diese Option für alle Slavegeräte aus, wenn Ihr Netzwerk einen Master und mehrere Slaves umfasst. Für "Ausgangszustand Empfangsleitung":
  - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 3).
  - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 2).
  - Wählen Sie "Vorspannung in Sperrichtung", um interne Vorspannung und Abschluss zu verwenden und die Kabelbrucherkennung für die Slaves zu aktivieren (Fall 1).

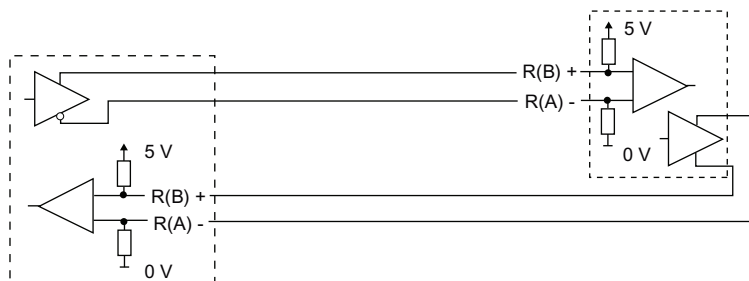
**Fall 1: RS422 mit Kabelbrucherkennung**

- Betriebsart: RS422
- Ausgangszustand Empfangsleitung: Vorspannung in Sperrichtung (Vorspannung mit  $R(A) > R(B) > 0 V$ )
- Kabelbruch: Kabelbrucherkennung aktiviert (Sender immer aktiv)



**Fall 2: RS422 ohne Kabelbrucherkennung, Vorspannung in Vorwärtsrichtung**

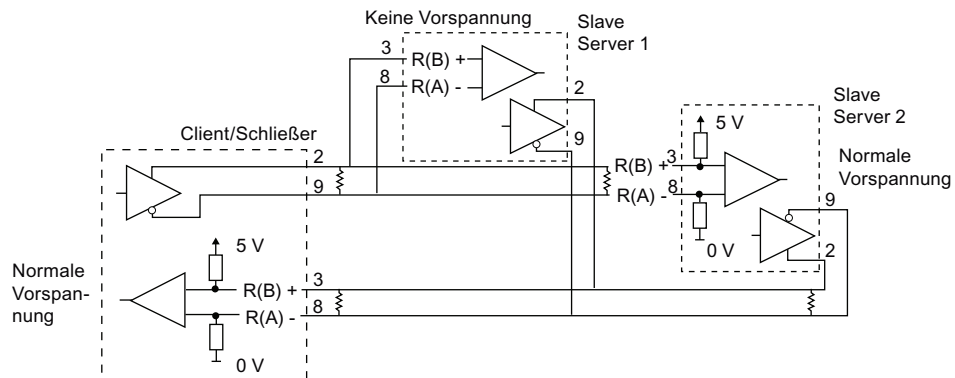
- Betriebsart: RS422
- Ausgangszustand Empfangsleitung: Vorspannung in Vorwärtsrichtung (Vorspannung mit  $R(B) > R(A) > 0 V$ )
- Kabelbruch: Keine Kabelbrucherkennung (Sender nur beim Senden aktiviert)



**Fall 3: RS422: Keine Kabelbrucherkennung, keine Vorspannung**

- Betriebsart: RS422
- Ausgangszustand Empfangsleitung: Keine Vorspannung
- Kabelbruch: Keine Kabelbrucherkennung (Sender nur beim Senden aktiviert)

Vorspannung und Abschluss werden vom Anwender an den Endknoten des Netzwerks bereitgestellt.



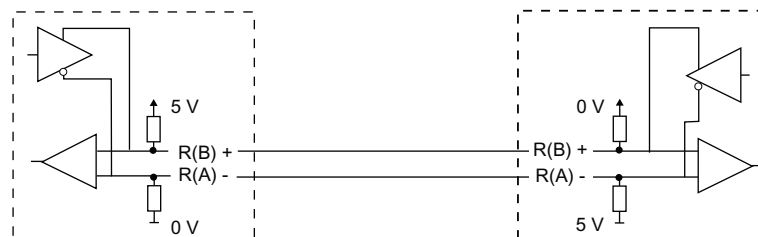
**RS485 konfigurieren**

Im RS485-Modus gibt es nur eine Betriebsart. Die verschiedenen Einstellungen für "Ausgangszustand Empfangsleitung" beziehen sich auf die im Folgenden dargestellten Fälle.

- Halbduplex (RS485), Zweidrahtmodus. Für "Ausgangszustand Empfangsleitung":
  - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 5).
  - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 4).

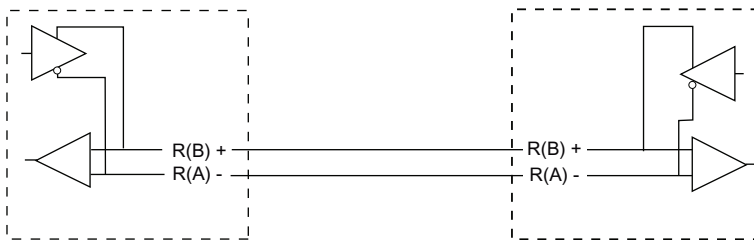
**Fall 4: RS485: Vorspannung in Vorwärtsrichtung**

- Betriebsart: RS485
- Ausgangszustand Empfangsleitung: Vorspannung in Vorwärtsrichtung (Vorspannung mit  $R(B) > R(A) > 0 V$ )



**Fall 5: RS485: Keine Vorspannung (externe Vorspannung)**

- Betriebsart: RS485
- Ausgangszustand Empfangsleitung: Keine Vorspannung (externe Vorspannung erforderlich)

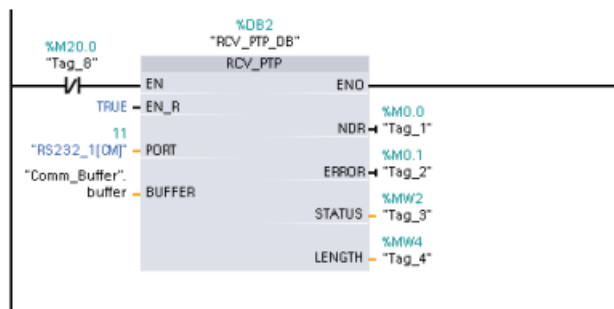


**12.3.5.3 STEP 7-Programm programmieren**

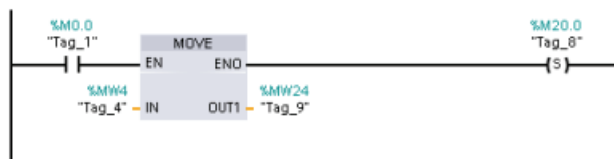
Das Beispielprogramm verwendet einen globalen Datenbaustein für den Kommunikationspuffer, eine Anweisung RCV\_PTP (Seite 614) für den Empfang von Daten vom Terminalemulator und eine Anweisung SEND\_PTP (Seite 610) zum Zurücksenden des Pufferechos an den Terminalemulator. Um das Beispiel zu programmieren, fügen Sie die Datenbausteinkonfiguration ein und programmieren OB1 wie im Folgenden beschrieben.

**Globaler Datenbaustein "Comm\_Buffer":** Erstellen Sie einen globalen Datenbaustein (DB) und nennen Sie ihn "Comm\_Buffer". Erstellen Sie einen Wert im Datenbaustein, nämlich "buffer", mit dem Datentyp "Array [0 .. 99] of byte".

**Netzwerk 1:** Aktivieren Sie die Anweisung RCV\_PTP immer, wenn SEND\_PTP nicht aktiv ist. Tag\_8 in MW20.0 zeigt in Netzwerk 4 an, wenn der Sendevorgang abgeschlossen ist und wenn das Kommunikationsmodul damit für den Meldungsempfang bereit ist.

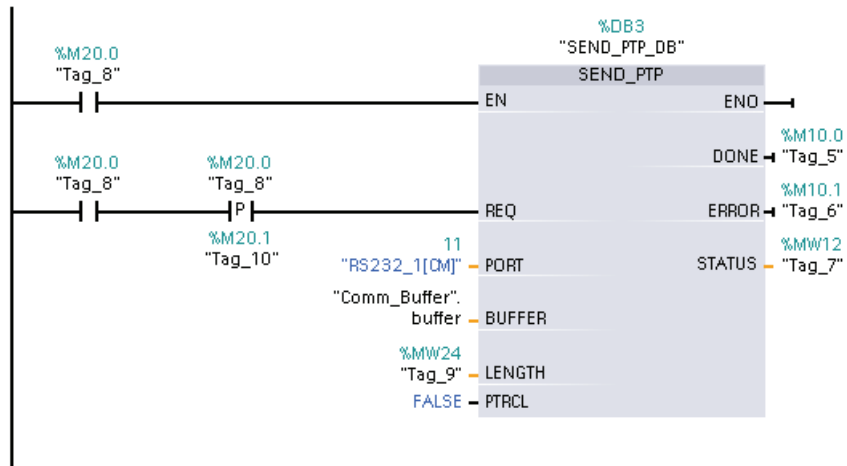


**Netzwerk 2:** Erstellen Sie mit dem von der Anweisung RCV\_PTP gesetzten NDR Wert (Tag\_1 in M0.0) eine Kopie von der empfangenen Anzahl Bytes und setzen Sie einen Merker (Tag\_8 in M20.0), um die Anweisung SEND\_PTP auszulösen.

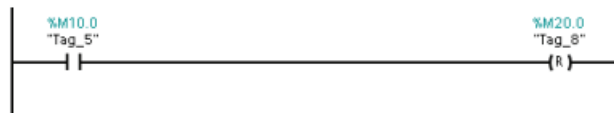




**Netzwerk 3:** Aktivieren Sie die Anweisung SEND\_PTP, wenn der Merker M20.0 gesetzt ist. Mit diesem Merker setzen Sie auch den Eingang REQ einen Zyklus lang auf WAHR. Der Eingang REQ teilt der Anweisung SEND\_PTP mit, dass eine neue Anforderung zu übertragen ist. Der Eingang REQ darf nur während einer Ausführung von SEND\_PTP auf WAHR gesetzt sein. Die Anweisung SEND\_PTP wird in jedem Zyklus ausgeführt, bis die Übertragung beendet ist. Die Übertragung ist beendet, wenn das letzte Byte der Meldung vom CM 1241 übertragen wurde. Wenn die Übertragung beendet ist, wird der Ausgang DONE (Tag\_5 in M10.0) während einer Ausführung von SEND\_PTP auf WAHR gesetzt.



**Netzwerk 4:** Überwachen Sie den Ausgang DONE von SEND\_PTP und setzen Sie den Übertragungsmerker (Tag\_8 in M20.0) zurück, wenn der Übertragungsvorgang beendet ist. Wenn der Übertragungsmerker zurückgesetzt wird, wird die Anweisung RCV\_PTP in Netzwerk 1 aktiviert, um die nächste Meldung zu empfangen.



#### 12.3.5.4 Terminalemulator konfigurieren

Sie müssen den Terminalemulator einrichten, um das Beispielprogramm auszuführen. Sie können nahezu jeden Terminalemulator an Ihrem PC verwenden, z. B. HyperTerminal. Achten Sie darauf, dass der Terminalemulator ausgeschaltet ist, bevor Sie die Einstellungen wie folgt ändern:

1. Legen Sie fest, dass der Terminalemulator den RS232-Anschluss am PC verwendet (normalerweise COM1).
2. Konfigurieren Sie den Port für 9600 Baud, 8 Datenbits, keine Parität, 1 Stoppbit und keine Flusskontrolle.
3. Ändern Sie die Einstellungen des Terminalemulators, um ein ANSI-Terminal zu emulieren.

4. Konfigurieren Sie die ASCII-Einrichtung des Terminalemulators so, dass nach jeder Zeile eine Zeilenschaltung gesendet wird (nachdem der Anwender die Eingabetaste drückt).
5. Geben Sie ein lokales Echo der Zeichen zurück, so dass der Terminalemulator anzeigt, was eingegeben wird.

### 12.3.5.5 Beispielprogramm ausführen

Um das Beispielprogramm auszuführen, gehen Sie folgendermaßen vor:

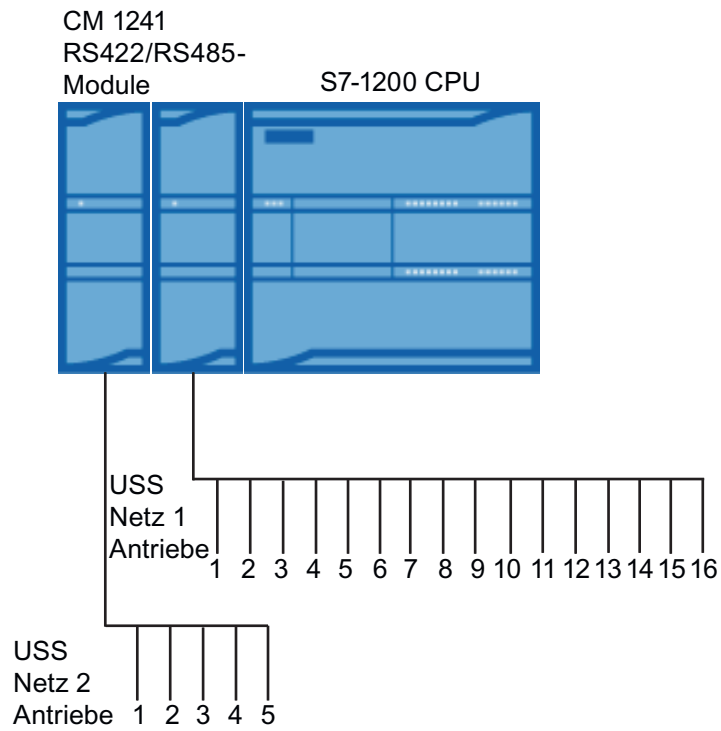
1. Laden Sie das STEP 7-Programm in die CPU und achten Sie darauf, dass diese im Betriebszustand RUN ist.
2. Klicken Sie im Terminalemulator auf die Schaltfläche zum Verbinden, um die Konfigurationsänderungen zu übernehmen, und öffnen Sie eine Terminalsitzung für das CM 1241.
3. Geben Sie am PC Zeichen ein und drücken Sie die Eingabetaste.

Der Terminalemulator sendet die Zeichen zum CM 1241 und zur CPU. Das CPU-Programm gibt dann ein Echo der Zeichen zum Terminalemulator zurück.

## 12.4 Kommunikation über die universelle serielle Schnittstelle (USS)

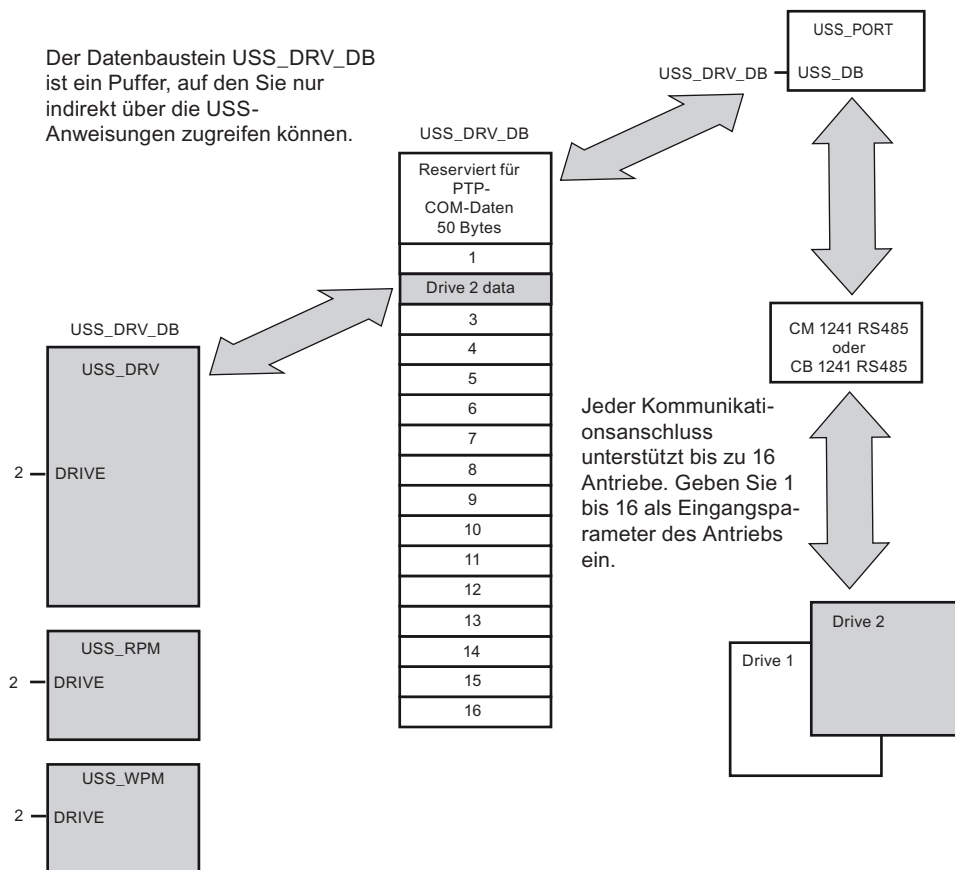
Die USS-Anweisungen steuern den Betrieb von Motorantrieben, die das Protokoll der universellen seriellen Schnittstelle (USS) unterstützen. Mit den USS-Anweisungen können Sie über RS485-Verbindungen mit mehreren Antrieben mit CM 1241 RS485-Kommunikationsmodulen oder einem CB 1241 RS485-Kommunikationsboard kommunizieren. In einer S7-1200 CPU können bis zu drei CM 1241 RS422/RS485-Module und ein CB 1241 RS485-Board eingebaut werden. Jeder RS485-Port kann bis zu sechzehn Antriebe betreiben.

Das USS-Protokoll nutzt ein Master/Slave-Netzwerk für die Kommunikation über einen seriellen Bus. Der Master verwendet einen Adressparameter, um eine Meldung an einen ausgewählten Slave zu senden. Ein Slave selbst kann niemals senden, ohne dafür zuvor eine Anforderung zu erhalten. Die direkte Meldungsübertragung zwischen den einzelnen Slaves ist nicht möglich. Die USS-Kommunikation funktioniert im Halbduplex-Betrieb. Die folgende USS-Abbildung zeigt ein Netzwerkdiagramm für eine Beispielanwendung eines Antriebs.



### 12.4.1 Voraussetzungen für den Einsatz des USS-Protokolls

Die vier USS-Anweisungen verwenden 1 FB und 3 FCs zur Unterstützung des USS-Protokolls. Für jedes USS-Netzwerk wird ein Instanz-Datenbaustein (DB) USS\_PORT verwendet. Der Instanz-Datenbaustein USS\_PORT enthält temporäre Speicher und Puffer für alle Antriebe in dem USS-Netzwerk. Die USS-Anweisungen nutzen die Informationen in diesem Datenbaustein gemeinsam.



Alle Antriebe (max. 16), die an einen RS485-Port angeschlossen sind, sind Teil desselben USS-Netzwerks. Alle Antriebe, die an einen anderen RS485-Port angeschlossen sind, sind Teil eines anderen USS-Netzwerks. Jedes USS-Netzwerk wird mithilfe eines eindeutigen Datenbausteins verwaltet. Alle Anweisungen, die zu einem USS-Netzwerk gehören, müssen diesen Datenbaustein gemeinsam nutzen. Dies umfasst alle Anweisungen USS\_DRV, USS\_PORT, USS\_RPM und USS\_WPM für die Steuerung aller Antriebe in einem USS-Netzwerk.

Die Anweisung USS\_DRV ist ein Funktionsbaustein (FB). Wenn Sie die Anweisung USS\_DRV in den Programmiereditor einfügen, werden Sie im Dialog "Aufrufoptionen" aufgefordert, einen DB für diesen FB zuzuweisen. Wenn es sich um die erste Anweisung USS\_DRV in diesem Programm für dieses USS-Netzwerk handelt, können Sie die DB-Standardzuweisung übernehmen (oder ggf. den Namen ändern), und der neue DB wird für Sie erstellt. Wenn es sich jedoch nicht um die erste Anweisung USS\_DRV für diesen Kanal handelt, müssen Sie im Dialog "Aufrufoptionen" in der Klappliste den DB auswählen, der diesem USS-Netzwerk bereits zuvor zugewiesen wurde.

Bei allen Anweisungen USS\_PORT, USS\_RPM und USS\_WPM handelt es sich um Funktionen (FCs). Wenn Sie diese FCs im Editor einfügen, wird kein DB zugewiesen. Stattdessen müssen Sie dem Eingang USS\_DB dieser Anweisungen den jeweiligen DB zuweisen. Doppelklicken Sie auf das Parameterfeld und klicken Sie dann auf das Symbol, um die verfügbaren DBs anzuzeigen.

Die Funktion USS\_PORT steuert die Kommunikation zwischen der CPU und den Antrieben über den Punkt-zu-Punkt(PtP)-RS485-Kommunikationsport. Bei jedem Aufruf dieser Funktion wird eine Kommunikation mit einem Antrieb bearbeitet. Ihr Programm muss diese Funktion schnell genug aufrufen, so dass die Antriebe keine Zeitüberschreitung melden. Diese Funktion kann aus dem Zyklus-OB des Hauptprogramms oder aus einem beliebigen Alarm-OB aufgerufen werden.

Typischerweise rufen Sie die Funktion USS\_PORT aus einem Weckalarm-OB auf. Die Zykluszeit des Weckalarm-OB setzen Sie auf ungefähr ein halbes Mindestaufrufintervall (beispielsweise sollte für die Kommunikation mit 1200 Baud eine Zykluszeit von maximal 350 ms verwendet werden).

Der Funktionsbaustein USS\_DRV gibt Ihrem Programm Zugriff auf einen angegebenen Antrieb im USS-Netzwerk. Seine Ein- und Ausgänge entsprechen den Zuständen und den Bedienfunktionen des Antriebs. Sind 16 Antriebe im Netzwerk vorhanden, so muss USS\_DRV in Ihrem Programm mindestens 16mal aufgerufen werden, also jeweils einmal für jeden Antrieb. Wie schnell diese Bausteine aufgerufen werden, hängt von der erforderlichen Geschwindigkeit für die Steuerung des Antriebsbetriebs ab.

Sie können den Funktionsbaustein USS\_DRV nur aus dem Zyklus-OB eines Hauptprogramms aufrufen.

#### VORSICHT

Rufen Sie USS\_DRV, USS\_RPM und USS\_WPM nur aus einem Zyklus-OB des Hauptprogramms auf. Die Funktion USS\_PORT kann aus einem beliebigen OB aufgerufen werden, üblicherweise wird sie aus einem Weckalarm-OB aufgerufen.

Verwenden Sie die Anweisungen USS\_DRV, USS\_RPM und USS\_WPM nicht in einem OB mit einer höheren Priorität als die entsprechende Anweisung USS\_PORT. Fügen Sie beispielsweise USS\_PORT nicht in das Hauptprogramm und USS\_RPM nicht in einen Weckalarm-OB ein. Wird die Unterbrechung der Ausführung von USS\_PORT nicht verhindert, kann es zu unerwarteten Fehlern kommen.

Mit den Funktionen USS\_RPM und USS\_WPM werden die Betriebsparameter des entfernten Antriebs gelesen und geschrieben. Diese Parameter steuern die interne Funktionsweise des Antriebs. Eine Definition dieser Parameter finden Sie im Handbuch des Antriebs. Ihr Programm kann eine beliebige Anzahl dieser Funktionen enthalten, es kann jedoch immer nur eine Lese- oder Schreibanforderung für einen Antrieb aktiv sein. Sie dürfen die Funktionen USS\_RPM und USS\_WPM nur aus dem Zyklus-OB eines Hauptprogramms aufrufen.

### Zeit für die Kommunikation mit dem Antrieb berechnen

Die Kommunikation mit dem Antrieb läuft asynchron zum Zyklus der S71200 ab. Die S7-1200 durchläuft üblicherweise mehrere Zyklen, bevor die Kommunikation mit einem Antrieb beendet ist.

12.4 Kommunikation über die universelle serielle Schnittstelle (USS)

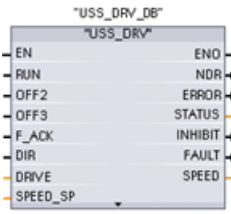

Das Intervall USS\_PORT ist die Zeit, die für eine Transaktion des Antriebs erforderlich ist. Die folgende Tabelle zeigt die Mindestintervalle für USS\_PORT für jede Baudrate der Kommunikation. Wenn Sie die Funktion USS\_PORT häufiger aufrufen, als es das USS\_PORT-Intervall vorgibt, wird die Anzahl der Transaktionen nicht erhöht. Das Timeout-Intervall des Antriebs ist die Zeitdauer, die für eine Transaktion zur Verfügung steht, wenn zur Fertigstellung der Transaktion aufgrund von Kommunikationsfehlern 3 Versuche nötig sind. Standardmäßig führt die Bibliothek für das USS-Protokoll bei jeder Transaktion bis zu 2 Wiederholungen durch.

Tabelle 12- 34 Zeitbedarf berechnen

Baudrate	Berechnetes Mindestintervall für Aufruf von USS_PORT (ms)	Intervall-Timeout für Antriebsmeldung pro Antrieb (ms)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

## 12.4.2 Operation USS\_DRV

Tabelle 12- 35 Anweisung USS\_DRV

KOP/FUP	SCL	Beschreibung
<p>Standardansicht</p>  <p>Erweiterte Ansicht</p> 	<pre>"USS_DRV_DB" (     RUN:=_bool_in_,     OFF2:=_bool_in_,     OFF3:=_bool_in_,     F_ACK:=_bool_in_,     DIR:=_bool_in_,     DRIVE:=_usint_in_,     PZD_LEN:=_usint_in_,     SPEED_SP:=_real_in_,     CTRL3:=_word_in_,     CTRL4:=_word_in_,     CTRL5:=_word_in_,     CTRL6:=_word_in_,     CTRL7:=_word_in_,     CTRL8:=_word_in_,     NDR=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_,     RUN_EN=&gt;_bool_out_,     D_DIR=&gt;_bool_out_,     INHIBIT=&gt;_bool_out_,     FAULT=&gt;_bool_out_,     SPEED=&gt;_real_out_,     STATUS1=&gt;_word_out_,     STATUS3=&gt;_word_out_,     STATUS4=&gt;_word_out_,     STATUS5=&gt;_word_out_,     STATUS6=&gt;_word_out_,     STATUS7=&gt;_word_out_,     STATUS8=&gt;_word_out_);</pre>	<p>Die Anweisung USS_DRV tauscht Daten mit einem Antrieb aus, indem Anfragemeldungen erzeugt und die Antwortmeldungen des Antriebs ausgewertet werden. Es sollte für jeden Antrieb ein eigener Funktionsbaustein verwendet werden, jedoch müssen alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, den gleichen Instanz-Datenbaustein verwenden. Sie müssen den DB-Namen eingeben, wenn Sie die erste Anweisung USS_DRV einfügen. Dann verweisen Sie auf diesen DB, der beim Einfügen der ersten Anweisung angelegt wurde.</p> <p>STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.</p>

<sup>1</sup> KOP und FUP: Erweitern Sie die Box, um alle Parameter anzuzeigen. Klicken Sie dazu auf den unteren Bereich der Box. Die Parameteranschlüsse, die grau dargestellt sind, sind optional, eine Parametrierung ist nicht erforderlich.

Tabelle 12- 36 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
RUN	IN	Bool
OFF2	IN	Bool

Startbit des Antriebs: Ist dieser Parameter WAHR, so ermöglicht dieser Eingang den Betrieb des Antriebs mit der voreingestellten Drehzahl. Wenn RUN im Betrieb des Antriebs nach Falsch wechselt, läuft der Motor bis zum Stillstand aus. Dieses Verhalten unterscheidet sich von der Abschaltung der Spannungsversorgung (OFF2) und vom Bremsen des Motors (OFF3).

Bit "Zum Stillstand auslaufen": Ist dieser Parameter FALSCH, so veranlasst dieses Bit das Auslaufen des Antriebs, ohne zu bremsen.

Parameter und Datentyp		Datentyp	Beschreibung
OFF3	IN	Bool	Schnelles Stoppbit: Ist dieser Parameter FALSCH, so verursacht dieses Bit einen schnellen Halt durch Abbremsen des Antriebs.
F_ACK	IN	Bool	Fehlerquittierungsbit: Mit diesem Bit wird das Fehlerbit eines Antriebs zurückgesetzt. Das Bit wird nach dem Löschen des Fehlers gesetzt und der Antrieb erkennt damit, dass der vorherige Fehler nicht mehr gemeldet werden muss.
DIR	IN	Bool	Richtungssteuerung des Antriebs: Dieses Bit wird gesetzt, wenn der Antrieb in Vorwärtsrichtung laufen soll (wenn SPEED_SP positiv ist).
DRIVE	IN	USInt	Adresse des Antriebs: Dieser Eingang ist die Adresse des USS-Antriebs. Der gültige Bereich liegt zwischen Antrieb 1 und Antrieb 16.
PZD_LEN	IN	USInt	Wortlänge: Dies ist die Anzahl der PZD-Datenwörter. Gültige Werte sind 2, 4, 6 oder 8 Wörter. Der Standardwert ist 2.
SPEED_SP	IN	Real	Drehzahlsollwert: Dies ist die Drehzahl des Antriebs prozentual zur konfigurierten Frequenz. Ein positiver Wert bedeutet, dass der Antrieb vorwärts läuft (wenn DIR wahr ist). Gültig ist der Bereich von 200,00 bis -200,00.
CTRL3	IN	Word	Steuerwort 3: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL4	IN	Word	Steuerwort 4: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL5	IN	Word	Steuerwort 5: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL6	IN	Word	Steuerwort 6: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL7	IN	Word	Steuerwort 7: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL8	IN	Word	Steuerwort 8: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
NDR	OUT	Bool	Neue Daten bereit: Ist dieser Parameter wahr, so meldet das Bit, dass am Ausgang Daten einer neuen Kommunikationsanforderung bereitstehen.
ERROR	OUT	Bool	Fehler aufgetreten: Wenn WAHR, weist dies darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_PORT gemeldet.
STATUS	OUT	Word	Der Zustandswert der Anforderung zeigt des Ergebnis des Zyklus an. Dies ist kein vom Antrieb ausgegebenes Zustandswort.
RUN_EN	OUT	Bool	Betrieb freigegeben: Dieses Bit meldet, ob der Antrieb läuft.
D_DIR	OUT	Bool	Antriebsrichtung: Dieses Bit meldet, ob der Antrieb vorwärts läuft.
INHIBIT	OUT	Bool	Antrieb gesperrt: Dieses Bit meldet den Zustand des Sperrbits für den Antrieb.



Parameter und Datentyp		Datentyp	Beschreibung
FAULT	OUT	Bool	Antriebsfehler: Dieses Bit meldet, dass im Antrieb ein Fehler aufgetreten ist. Sie müssen die Störung beheben und Bit F_ACK setzen, um dieses Bit zu löschen.
SPEED	OUT	Real	Istwert Antriebsdrehzahl (skalierter Wert von Zustandswort 2 des Antriebs): Dies ist die Drehzahl des Antriebs prozentual zur konfigurierten Drehzahl.
STATUS1	OUT	Word	Zustandswort 1 des Antriebs: Dieser Wert enthält feste Zustandsbits eines Antriebs.
STATUS3	OUT	Word	Zustandswort 3 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS4	OUT	Word	Zustandswort 4 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS5	OUT	Word	Zustandswort 5 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS6	OUT	Word	Zustandswort 6 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS7	OUT	Word	Zustandswort 7 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS8	OUT	Word	Zustandswort 8 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.

Wenn die erste Ausführung von USS\_DRV erfolgt, wird der von der USS-Adresse (Parameter DRIVE) angegebene Antrieb im Instanz-DB initialisiert. Nach dieser Initialisierung können nachfolgende Anweisungen USS\_PORT die Kommunikation mit dem Antrieb an dieser Antriebsnummer beginnen.

Wenn Sie die Antriebsnummer ändern, muss die CPU zunächst in STOP und dann wieder in RUN versetzt werden, damit der Instanz-DB initialisiert wird. Die Eingangsparameter werden im USS-Sendepuffer konfiguriert und die Ausgänge werden, sofern vorhanden, aus einem "vorherigen" gültigen Antwortpuffer gelesen. Während der Ausführung der Anweisung USS\_DRV findet keine Datenübertragung statt. Nach der Ausführung von USS\_PORT kommunizieren die Antriebe. USS\_DRV konfiguriert nur die zu sendenden Meldungen und wertet Daten aus, die in einer vorherigen Anforderung empfangen wurden.

Sie können die Drehrichtung des Antriebs entweder über den Eingang DIR (Bool) oder über das Vorzeichen (positiv oder negativ) am Eingang SPEED\_SP (Real) steuern. Die folgende Tabelle erläutert, wie diese Eingänge zusammen funktionieren, um die Drehrichtung des Antriebs zu bestimmen, vorausgesetzt der Motor dreht vorwärts.

Tabelle 12- 37 Interaktion der Parameter SPEED\_SP und DIR

SPEED_SP	DIR	Drehrichtung des Antriebs
Wert > 0	0	Rückwärts
Wert > 0	1	Vorwärts
Wert < 0	0	Vorwärts
Wert < 0	1	Rückwärts

### 12.4.3 Operation USS\_PORT

Tabelle 12- 38 Anweisung USS\_PORT

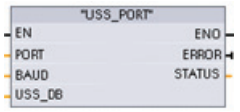
KOP/FUP	SCL	Beschreibung
	<pre>USS_PORT (     PORT:= _uint_in_,     BAUD:= _dint_in_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_,     USS DB:= fbtref inout );</pre>	<p>Die Anweisung USS_PORT bearbeitet die Kommunikation über ein USS-Netzwerk.</p>

Tabelle 12- 39 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
PORT	IN	Port	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen.
BAUD	IN	DInt	Die Baudrate für die USS-Kommunikation.
USS_DB	INOUT	USS_BASE	Der Name des Instanz-DBs, der erstellt und initialisiert wird, wenn eine Anweisung USS_DRV in Ihr Programm eingefügt wird.
ERROR	OUT	Bool	Wenn WAHR, weist dieser Ausgang darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist.
STATUS	OUT	Word	Der Zustandswert der Anforderung zeigt des Ergebnis des Zyklus oder der Initialisierung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

Normalerweise ist nur eine Anweisung USS\_PORT pro PtP-Kommunikationsport im Programm vorhanden und jeder Aufruf dieser Anweisung steuert eine Übertragung zu oder von einem einzigen Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Instanz-DB nutzen.

Ihr Programm muss die Anweisung USS\_PORT so oft ausführen, dass kein Timeout im Antrieb auftritt. Die Anweisung USS\_PORT wird üblicherweise aus einem Weckalarm-OB aufgerufen, um Antriebs-Timeouts zu verhindern und die letzten USS-Datenaktualisierungen für Aufrufe von USS\_DRV verfügbar zu haben.

## 12.4.4 Operation USS\_RPM

Tabelle 12- 40 Anweisung USS\_RPM

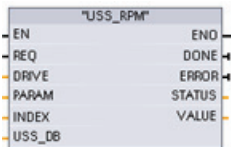
KOP/FUP	SCL	Beschreibung
	<pre>USS_RPM(REQ:=_bool_in_,         DRIVE:=_usint_in_,         PARAM:=_uint_in_,         INDEX:=_uint_in_,         DONE=&gt;_bool_out_,         ERROR=&gt;_bool_out_,         STATUS=&gt;_word_out_,         VALUE=&gt;_variant_out_,         USS_DB:=_fbtref_inout_);</pre>	<p>Die Anweisung USS_RPM liest einen Parameter aus einem Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Datenbaustein nutzen. USS_RPM muss aus einem Zyklus-OB des Hauptprogramms aufgerufen werden.</p>

Tabelle 12- 41 Datentypen für die Parameter

Parametertyp	Datentyp	Beschreibung	
REQ	IN	Bool	Sendeanforderung: Ist REQ WAHR, so wird eine neue Leseanforderung benötigt. Dies wird ignoriert, wenn die Anforderung für diesen Parameter bereits ansteht.
DRIVE	IN	USInt	Adresse des Antriebs: DRIVE ist die Adresse des USS-Antriebs. Der gültige Bereich liegt zwischen Antrieb 1 und Antrieb 16.
PARAM	IN	UInt	Parameternummer: PARAM gibt an, welcher Antriebsparameter geschrieben wird. Der Bereich für diesen Parameter liegt zwischen 0 und 2047. Bei einigen Antrieben kann das höchstwertige Byte auf PARAM-Werte größer als 2047 zugreifen. Weitere Informationen für den Zugriff auf einen erweiterten Bereich finden Sie im Handbuch zu Ihrem Antrieb.
INDEX	IN	UInt	Parameterindex: INDEX gibt an, in welchen Antriebsparameterindex geschrieben werden soll. Es handelt sich um einen 16-Bit-Wert, bei dem das niederwertigste Byte der tatsächliche Indexwert ist, mit einem Bereich von (0 bis 255). Das höchstwertige Byte kann ebenfalls von dem Antrieb verwendet werden und ist antriebsspezifisch. Weitere Informationen finden Sie im Handbuch Ihres Antriebs.
USS_DB	INOUT	USS_BASE	Der Name des Instanz-DBs, der erstellt und initialisiert wird, wenn eine Anweisung USS_DRV in Ihr Programm eingefügt wird.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Dies ist der Wert des Parameters, der gelesen wurde und er ist nur gültig, wenn das Bit DONE wahr ist.
DONE <sup>1</sup>	OUT	Bool	Ist dieser Parameter WAHR, so steht am Ausgang VALUE der zuvor angeforderte Wert des Leseparameters an. Dieses Bit wird gesetzt, wenn die Anweisung USS_DRV die Leseantwort des Antriebs erkennt. Dieses Bit wird zurückgesetzt, wenn: Sie die Antwortdaten über eine andere Abfrage USS_RPM anfordern oder beim zweiten der nächsten beiden Aufrufe von USS_DRV

Parametertyp		Datentyp	Beschreibung
ERROR	OUT	Bool	Fehler aufgetreten: Wenn WAHR, weist ERROR darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_PORT gemeldet.
STATUS	OUT	Word	STATUS gibt das Ergebnis der Leseanforderung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

<sup>1</sup> Das Bit DONE weist darauf hin, dass gültige Daten aus dem referenzierten Motorantrieb ausgelesen und an die CPU geliefert wurden. Es weist nicht darauf hin, dass die USS-Bibliothek in der Lage ist, sofort einen weiteren Parameter auszulesen. Eine leere PKW-Anforderung muss an den Motorantrieb gesendet und auch von der Anweisung quittiert werden, bevor der Parameterkanal zur Verwendung durch den jeweiligen Antrieb frei wird. Der sofortige Aufruf von USS\_RPM oder USS\_WPM FC für den spezifischen Motorantrieb führt zu dem Fehler 0x818A.

### 12.4.5 Operation USS\_WPM

**Hinweis**

**EEPROM-Schreibanweisungen (für den EEPROM in einem USS-Antrieb)**

Übertreiben Sie die Verwendung der EEPROM-Schreiboperation nicht. Halten Sie die Anzahl der EEPROM-Schreiboperationen möglichst gering, um die Lebensdauer des EEPROM zu verlängern.

Tabelle 12- 42 Anweisung USS\_WPM

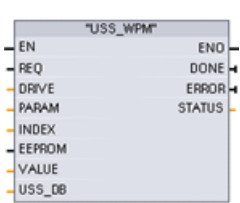
KOP/FUP	SCL	Beschreibung
	<pre> USS_WPM (REQ:=_bool_in_,           DRIVE:=_usint_in_,           PARAM:=_uint_in_,           INDEX:=_uint_in_,           EEPROM:=_bool_in_,           VALUE:=_variant_in_,           DONE=&gt;_bool_out_,           ERROR=&gt;_bool_out_,           STATUS=&gt;_word_out_,           USS_DB:=_fbtref inout );         </pre>	<p>Die Anweisung USS_WPM ändert einen Parameter im Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Datenbaustein nutzen.</p> <p>USS_WPM muss aus dem Zyklus-OB eines Hauptprogramms aufgerufen werden.</p>

Tabelle 12- 43 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Sendeanforderung: Ist REQ WAHR, so wird eine neue Schreibanforderung benötigt. Dies wird ignoriert, wenn die Anforderung für diesen Parameter bereits ansteht.
DRIVE	IN	USInt	Adresse des Antriebs: DRIVE ist die Adresse des USS-Antriebs. Der gültige Bereich liegt zwischen Antrieb 1 und Antrieb 16.
PARAM	IN	UInt	Parameternummer: PARAM gibt an, welcher Antriebsparameter geschrieben wird. Der Bereich für diesen Parameter liegt zwischen 0 und 2047. Bei einigen Antrieben kann das höchstwertige Byte auf PARAM-Werte größer als 2047 zugreifen. Weitere Informationen für den Zugriff auf einen erweiterten Bereich finden Sie im Handbuch zu Ihrem Antrieb.
INDEX	IN	UInt	Parameterindex: INDEX gibt an, in welchen Antriebsparameterindex geschrieben werden soll. Es handelt sich um einen 16-Bit-Wert, bei dem das niederwertigste Byte der tatsächliche Indexwert ist, mit einem Bereich von (0 bis 255). Das höchstwertige Byte kann ebenfalls von dem Antrieb verwendet werden und ist antriebsspezifisch. Weitere Informationen finden Sie im Handbuch Ihres Antriebs.
EEPROM	IN	Bool	Im EEPROM des Antriebs speichern: Wenn WAHR, wird die Transaktion eines Parameters zum Schreiben in den Antrieb im EEPROM des Antriebs gespeichert. Wenn FALSCH, so wird der geschriebene Wert nur temporär gespeichert und geht beim nächsten Einschalten des Antriebs verloren.
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real	Wert des Parameters, in den geschrieben werden soll. Er muss beim Zustandswechsel von REQ gültig sein.
USS_DB	INOUT	USS_BASE	Der Name des Instanz-DBs, der erstellt und initialisiert wird, wenn eine Anweisung USS_DRV in Ihr Programm eingefügt wird.
DONE <sup>1</sup>	OUT	Bool	Wenn WAHR, weist DONE darauf hin, dass der Eingang VALUE in den Antrieb geschrieben wurde. Dieses Bit wird gesetzt, wenn die Anweisung USS_DRV die Schreibantwort des Antriebs erkennt. Dieses Bit wird zurückgesetzt, entweder wenn Sie die Antwortdaten über eine weitere Abfrage USS_WPM anfordern, oder beim zweiten der nächsten beiden Aufrufe von USS_DRV.
ERROR	OUT	Bool	Wenn WAHR, weist ERROR darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_PORT gemeldet.
STATUS	OUT	Word	STATUS gibt das Ergebnis der Schreibanforderung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

<sup>1</sup> Das Bit DONE weist darauf hin, dass gültige Daten aus dem referenzierten Motorantrieb ausgelesen und an die CPU geliefert wurden. Es weist nicht darauf hin, dass die USS-Bibliothek in der Lage ist, sofort einen weiteren Parameter auszulesen. Eine leere PKW-Anforderung muss an den Motorantrieb gesendet und auch von der Anweisung quittiert werden, bevor der Parameterkanal zur Verwendung durch den jeweiligen Antrieb frei wird. Der sofortige Aufruf von USS\_RPM oder USS\_WPM FC für den spezifischen Motorantrieb führt zu dem Fehler 0x818A.

### 12.4.6 USS-Zustandscodes

Statuscodes der USS-Anweisung werden im Ausgang STATUS der USS-Funktionen ausgegeben.

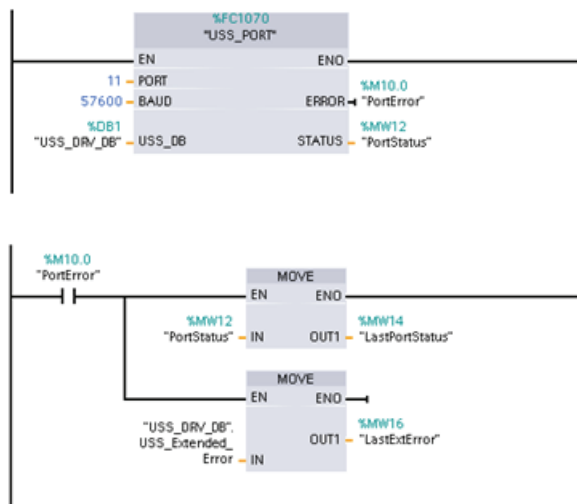
Tabelle 12- 44 STATUS-Codes <sup>1</sup>

STATUS (W#16#....)	Beschreibung
0000	Kein Fehler
8180	Die Länge der Antwort des Antriebs entsprach nicht den vom Antrieb empfangenen Zeichen. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8181	Der Parameter VALUE gehört nicht zum Datentyp Wort, Real oder Doppelwort.
8182	Der Anwender hat einen Parameterwert vom Typ Wort eingegeben und die Antwort vom Antrieb im Format Doppelwort oder Real empfangen.
8183	Der Anwender hat einen Parameterwert vom Typ Doppelwort oder Real eingegeben und die Antwort vom Antrieb im Format Wort empfangen.
8184	Das Antworttelegramm des Antriebs hatte eine falsche Prüfsumme. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8185	Unzulässige Antriebsadresse (gültiger Adressbereich für Antriebe: 1 bis16)
8186	Der Drehzahlsollwert liegt außerhalb des gültigen Bereichs (gültiger Sollwertbereich für die Drehzahl: - 200 % bis 200 %).
8187	Die falsche Antriebsnummer hat auf die gesendete Anforderung geantwortet. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8188	Unzulässige PZD-Wortlänge angegeben (gültiger Bereich = 2, 4, 6 oder 8 Wörter)
8189	Unzulässige Baudrate angegeben.
818A	Der Anforderungskanal für Parameter wird von einer anderen Anforderung für diesen Antrieb verwendet.
818B	Der Antrieb hat nicht auf Anforderungen und Wiederholungen reagiert. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
818C	Der Antrieb hat einen erweiterten Fehler zu einer Parameteranforderung ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
818D	Der Antrieb hat einen Fehler "Unzulässiger Zugriff" zu einer Parameteranforderung ausgegeben. Weitere Informationen dazu, weshalb der Parameterzugriff begrenzt sein kann, finden Sie im Handbuch zu Ihrem Antrieb.
818E	Der Antrieb wurde nicht initialisiert. Dieser Fehlercode wird an USS_RPM oder USS_WPM ausgegeben, wenn die Anweisung USS_DRV nicht mindestens einmal für diesen Antrieb aufgerufen wurde. Dadurch wird verhindert, dass die Initialisierung im ersten Zyklus von USS_DRV eine anstehende Anforderung zum Lesen oder Schreiben von Parametern überschreibt, weil dabei der Antrieb als neuer Eintrag initialisiert wird. Um diesen Fehler zu beheben, rufen Sie die Anweisung USS_DRV für diesen Antrieb auf.
80Ax-80Fx	Spezifische Fehler, die von den von der USS-Bibliothek aufgerufenen FBs für die PtP-Kommunikation zurückgegeben werden - Diese Fehlercodes werden von der USS-Bibliothek nicht verändert und sind in den Beschreibungen der PtP-Anweisung definiert.

<sup>1</sup> Neben den oben aufgeführten Fehlern der USS-Anweisungen können auch die zugrunde liegenden PtP-Kommunikationsanweisungen (Seite 600) Fehler zurückgeben.

Für verschiedene STATUS-Codes werden weitere Informationen in der Variable "USS\_Extended\_Error" des Instanz-DB USS\_DRV zur Verfügung gestellt. Für die STATUS-Codes hexadezimal 8180, 8184, 8187 und 818B, enthält USS\_Extended\_Error die Antriebsnummer des Antriebs, bei dem der Kommunikationsfehler auftrat. Für die STATUS-Codes hexadezimal 818C enthält USS\_Extended\_Error einen Antriebsfehlercode, der bei der Verwendung einer Anweisung USS\_RPM oder USS\_WPM vom Antrieb ausgegeben wird.

Kommunikationsfehler (STATUS = 16#818B) werden nur bei der Anweisung USS\_PORT und nicht bei der Anweisung USS\_DRV gemeldet. Beispiel: Wenn das Netzwerk nicht ordnungsgemäß beendet wird, kann ein Antrieb in RUN wechseln, doch die Anweisung USS\_DRV zeigt an allen Ausgangsparametern 0 an. In diesem Fall können Sie den Kommunikationsfehler nur über die Anweisung USS\_PORT erkennen. Weil dieser Fehler nur einen Zyklus lang sichtbar ist, müssen Sie entsprechende Erfassungslgik einfügen. Dies wird in dem folgenden Beispiel dargestellt. In diesem Beispiel werden, wenn das Fehlerbit der Anweisung USS\_PORT WAHR ist, die Werte STATUS und USS\_Extended\_Error im Speicherbereich der Merker abgelegt. Die Antriebsnummer wird in der Variable USS\_Extended\_Error abgelegt, wenn der Wert des STATUS-Codes hexadezimal 8180, 8184, 8187 oder 818B ist.



**Netzwerk 1** Der Portzustand "PortStatus" und die erweiterten Fehlercodewerte "USS\_DRV\_DB".USS\_Extended\_Error sind nur einen Programmzyklus lang gültig. Die Werte müssen zur späteren Bearbeitung erfasst werden.

**Netzwerk 2** Der "PortError"-Kontakt löst die Speicherung des "PortStatus"-Werts in "LastPortStatus" und des "USS\_DRV\_DB".USS\_Extended\_Error-Werts in "LastExtError" aus.

USS-Antriebe unterstützen Lese- und Schreibzugriff auf die internen Parameter eines Antriebs. Diese Funktion ermöglicht die dezentrale Steuerung und Konfiguration des Antriebs. Zugriffe der Antriebsparameter können aufgrund von Fehlern wie Wert außerhalb des Bereichs oder unzulässige Anforderungen in der aktuellen Betriebsart des Antriebs fehlschlagen. Der Antrieb erzeugt einen Fehlercode, der in der Variablen "USS\_Extended\_Error" ausgegeben wird. Dieser Fehlercode gilt nur für die letzte Ausführung der Anweisung USS\_RPM oder USS\_WPM. Der Fehlercode des Antriebs wird in der Variablen "USS\_Extended\_Error" abgelegt, wenn der Wert von STATUS code hexadezimal 818C ist. Der Fehlercode von "USS\_Extended\_Error" richtet sich nach der Variante des Antriebs. Eine Beschreibung der erweiterten Fehlercodes von Lese- und Schreibfunktionen für Parameter finden Sie im Handbuch des Antriebs.

## 12.4.7 Allgemeine Informationen zur Antriebseinrichtung

### Allgemeine Voraussetzungen für die Antriebseinrichtung

- Für die Antriebe muss die Verwendung von 4 PKW-Wörtern eingerichtet werden.
- Die Antriebe können für 2, 4, 6 oder 8 PZD-Wörter konfiguriert werden.
- Die Anzahl der PZD-Wörter im Antrieb muss dem Eingang PZD\_LEN der Anweisung USS\_DRV des Antriebs entsprechen.
- Die Baudrate aller Antriebe muss dem Eingang BAUD der Anweisung USS\_PORT entsprechen.
- Der Antrieb muss für die Fernsteuerung eingerichtet werden.
- Für den Frequenzsollwert an der COM-Verbindung des Antriebs muss USS festgelegt werden.
- Für die Antriebsadresse muss 1 bis 16 festgelegt sein. Diese Adresse muss dem Eingang DRIVE am Baustein USS\_DRV des Antriebs entsprechen.
- Für die Richtungssteuerung des Antriebs muss die Verwendung der Polarität des Antriebssollwerts eingerichtet werden.
- Das RS485-Netzwerk muss ordnungsgemäß abgeschlossen sein.

### MicroMaster-Antrieb anschließen

Diese Informationen zu SIEMENS MicroMaster-Antriebe dienen als Beispiel. Bei anderen Antrieben finden Sie die Einrichtungsanleitung im Handbuch des Antriebs.

Wenn Sie einen MicroMaster-Antrieb der Serie 4 (MM4) anschließen möchten, stecken Sie die Enden des RS485-Kabels in die beiden schraubenlosen Druckklemmen für den USS-Betrieb. Die S7-1200 kann mit dem herkömmlichen PROFIBUS-Kabel und den Steckverbindern an den MicroMaster-Antrieb angeschlossen werden.

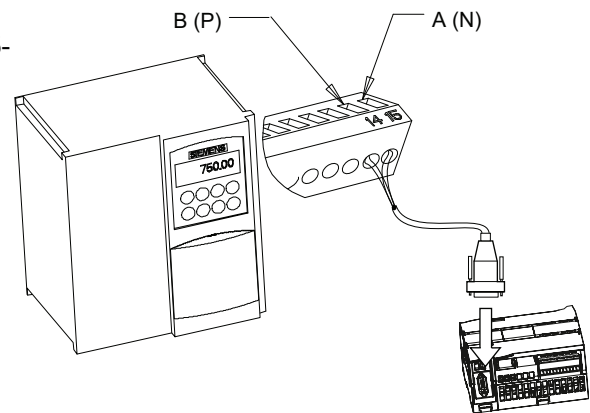
#### VORSICHT

**Wenn Sie Geräte miteinander verbinden, die nicht die gleiche Bezugsspannung haben, kann dies unerwünschte Ströme im Verbindungskabel hervorrufen.**

Diese unerwünschten Ströme können Kommunikationsfehler verursachen oder Sachschaden in den Geräten hervorrufen. Stellen Sie sicher, dass alle Geräte, die Sie über ein Kommunikationskabel miteinander verbinden, entweder den gleichen Bezugsleiter im Stromkreis haben oder elektrisch getrennt sind, damit keine unerwünschten Ströme auftreten. Die Schirmung muss mit Masse oder Pin 1 des 9-poligen Steckverbinders verbunden werden. Die Verdrahtungsklemme 2-0V des MicroMaster-Antriebs sollte mit Masse verbunden werden.



Stecken Sie die beiden Drähte am gegenüberliegenden Ende des RS485-Kabels in den Klemmenblock des MM4-Antriebs. Zum Herstellen der Kabelverbindung am MM4-Antrieb entfernen Sie die Abdeckung(en) des Antriebs, damit Sie Zugriff auf die Klemmenblöcke haben. Im Benutzerhandbuch des MM4-Antriebs finden Sie eine ausführliche Beschreibung, wie Sie die Abdeckung(en) an Ihrem spezifischen Antrieb entfernen.



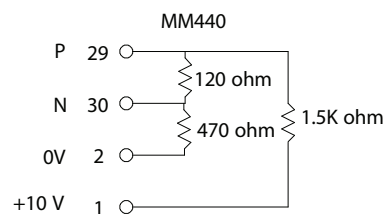
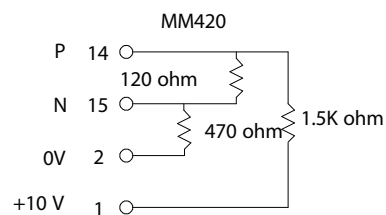
Die Anschlüsse am Klemmenblock sind durchnummeriert. Verwenden Sie auf der Seite der S7-1200 einen PROFIBUS-Anschlussstecker und schließen Sie die Klemme A des Kabels an Klemme 15 des Antriebs (bei einem MM420) oder an Klemme 30 des Antriebs (bei einem MM440) an. Schließen Sie die Klemme B (P) A (N) des Kabelverbinders an Klemme 14 (bei einem MM420) oder an Klemme 29 (bei einem MM440) an.

Handelt es sich bei der S71200 um einen abschließenden Teilnehmer im Netz, oder handelt es sich um eine Punkt-zu-Punkt-Verbindung, müssen Sie die Klemmen A1 und B1 (nicht A2 und B2) des Steckverbinders verwenden, weil diese Klemmen die Abschlusseinstellungen ermöglichen (z. B. beim DP-Steckverbinder 6ES7 972--0BA40--0X40).

### VORSICHT

Achten Sie darauf, dass Sie die Abdeckungen des Antriebs sorgfältig wieder einbauen, bevor Sie das Gerät mit Spannung versorgen.

Ist der Antrieb als abschließender Teilnehmer im Netz konfiguriert, dann müssen außerdem Abschlusswiderstände mit den entsprechenden Klemmen verdrahtet werden. Diese Abbildung zeigt Beispiele für die Anschlüsse des MM4-Antriebs für den Abschluss.



**Einrichten des MM4-Antriebs**

Bevor Sie einen Antrieb an die S7-1200 anschließen, müssen Sie sicherstellen, dass der Antrieb über folgende Systemparameter verfügt. Sie stellen die Parameter mit der Tastatur des Antriebs ein:

1. Setzen Sie den Antrieb auf die Werkseinstellungen zurück (optional).	P0010 = 30 P0970 = 1
Wenn Sie Schritt 1 überspringen, müssen Sie darauf achten, dass diese Parameter auf die angegebenen Werte gesetzt sind:	USS PZD-Länge = P2012 Index 0 = (2, 4, 6 oder 8) USS PKW-Länge = P2013 Index 0 = 4
2. Aktivieren Sie den Lese-/Schreibzugriff für alle Parameter (Expertenmodus).	P0003 = 3
3. Prüfen Sie die Motoreinstellungen Ihres Antriebs. Die Einstellungen richten sich nach dem jeweiligen Motor. Damit Sie die Parameter P304, P305, P307, P310 und P311 einstellen können, müssen Sie zunächst den Parameter P010 auf 1 setzen (Modus für Schnellinbetriebsetzung). Nachdem Sie die Parameter eingestellt haben, setzen Sie den Parameter P010 auf 0. Die Parameter P304, P305, P307, P310 und P311 können nur im Modus für Schnellinbetriebsetzung geändert werden.	P0304 = Motornennspannung (V) P0305 = Motornennstrom (A) P0307 = Motornennleistung (W) P0310 = Motornennfrequenz (Hz) P0311 = Motornennndrehzahl
4. Stellen Sie den lokalen/entfernten Modus ein.	P0700 Index 0 = 5
5. Stellen Sie den Frequenzsollwert der COM-Verbindung auf USS ein.	P1000 Index 0 = 5
6. Hochlaufzeit (optional) Zeit in Sekunden, die der Motor braucht, um auf maximale Frequenz zu beschleunigen.	P1120 = (0 bis 650,00)
7. Auslaufzeit (optional) Zeit in Sekunden, die der Motor braucht, um bis zum vollständigen Halt zu verzögern.	P1121 = (0 bis 650,00)
8. Stellen Sie die Referenzfrequenz der seriellen Verbindung ein:	P2000 = (1 bis 650 Hz)
9. Stellen Sie die USS-Normalisierung ein:	P2009 Index 0 = 0
10. Stellen Sie die Baudrate der seriellen Schnittstelle RS485 ein:	P2010 Index 0 = 4 (2400 Baud) 5 (4800 Baud) 6 (9600 Baud) 7 (19200 Baud) 8 (38400 Baud) 9 (57600 Baud) 12 (115200 Baud)
11. Geben Sie die Slave-Adresse ein. Jeder Antrieb (maximal 31) kann über den Bus betrieben werden.	P2011 Index 0 = (0 bis 31)
12. Legen Sie den Timeout-Wert für die serielle Verbindung fest. Dies ist der maximal zulässige Zeitraum zwischen zwei eingehenden Datentelegrammen. Diese Funktion schaltet den Inverter bei Kommunikationsausfall aus. Die Zeit wird gemessen, nachdem ein gültiges Telegramm empfangen wurde. Wird kein weiteres Datentelegramm innerhalb der angegebenen Zeit empfangen, schaltet der Inverter ab und zeigt Fehlercode F0070 an. Wenn Sie den Wert auf Null setzen, wird die Steuerung ausgeschaltet.	P2014 Index 0 = (0 bis 65.535 ms) 0 = Timeout deaktiviert
13. Übertragen Sie die Daten vom RAM zum EEPROM:	P0971 = 1 (Übertragung starten). Speichern Sie die Änderungen an den Parametereinstellungen im EEPROM.

## 12.5 Modbus-Kommunikation

### 12.5.1 Überblick über die Modbus RTU- und TCP-Kommunikation

#### Modbus-Funktionscodes

- Eine CPU, die als Modbus RTU-Master (oder Modbus TCP-Client) betrieben wird, kann Daten und E/A-Zustände in einem dezentralen Modbus RTU-Slave (bzw. Modbus TCP-Server) lesen und schreiben. Dezentrale Daten können vom Anwenderprogramm gelesen und verarbeitet werden.
- Eine CPU, die als Modbus RTU-Slave (oder Modbus TCP-Server) betrieben wird, ermöglicht es einem übergeordneten Gerät, Daten und E/A-Zustände in einer dezentralen CPU zu lesen und zu schreiben. Das übergeordnete Gerät kann neue Werte in den Speicher einer dezentralen CPU schreiben, die vom Anwenderprogramm verarbeitet werden können.

Tabelle 12- 45 Funktionen zum Lesen von Daten: Dezentrale E/A und Programmdaten lesen

Modbus-Funktionscode	Funktionen zum Lesen des Slave (Server) - Standardadressierung
01	Ausgangsbits lesen: 1 bis 2000 Bits pro Anforderung
02	Eingangsbits lesen: 1 bis 2000 Bits pro Anforderung
03	Halteregister lesen: 1 bis 125 Bits pro Anforderung
04	Eingangswörter lesen: 1 bis 125 Bits pro Anforderung

Tabelle 12- 46 Funktionen zum Schreiben von Daten: Dezentrale E/A schreiben und Programmdaten ändern

Modbus-Funktionscode	Funktionen zum Schreiben in den Slave (Server) - Standardadressierung
05	Ein Ausgangsbit schreiben: 1 Bit pro Anforderung
06	Ein Halteregister schreiben: 1 Wort pro Anforderung
15	Ein oder mehrere Ausgangsbits schreiben: 1 bis 1968 Bits pro Anforderung
16	Ein oder mehrere Halteregister schreiben: 1 bis 123 Bits pro Anforderung

- Die Modbus-Funktionscodes 08 und 11 bieten Diagnoseinformationen für die Kommunikation des Slavegeräts.
- Modbus-Funktionscode 0 sendet eine Broadcast-Meldung an alle Slaves (ohne Slaveantwort). Die Broadcast-Funktion ist bei Modbus TCP nicht verfügbar, weil die Kommunikation verbindungs-basiert abläuft.

Tabelle 12- 47 Stationsadressen im Modbus-Netzwerk

Station		Adresse
RTU-Station	Standardstationsadresse	1 bis 247
	Erweiterte Stationsadresse	1 bis 65535
TCP-Station	Stationsadresse	IP-Adresse und Portnummer

### Modbus-Speicheradressen

Die tatsächlich verfügbare Anzahl von Modbus-Speicheradressen richtet sich nach der CPU-Variante, nach dem verfügbaren Arbeitsspeicher und danach, wie viel CPU-Speicher durch andere Programmdaten belegt ist. Die folgende Tabelle zeigt den Nennwert des Adressbereichs.

Tabelle 12- 48 Modbus-Speicheradressen

Station		Adressbereich
RTU-Station	Adresse im Standardspeicher	10 K
	Adresse im erweiterten Speicher	64 K
TCP-Station	Adresse im Standardspeicher	10 K

### Modbus RTU-Kommunikation

Modbus RTU (Remote Terminal Unit) ist ein Standardprotokoll für die Kommunikation im Netzwerk und verwendet die elektrische RS232- oder RS485-Verbindung für die serielle Datenübertragung zwischen Modbus-Geräten im Netzwerk. Eine CPU mit einem RS232 oder RS485 CM oder einem RS485 CB können Sie um PtP-Netzwerkports (Punkt zu Punkt) erweitern.

Modbus RTU nutzt ein Master/Slave-Netzwerk, in dem die gesamte Kommunikation von einem einzigen Master-Gerät ausgelöst wird, während die Slaves lediglich auf die Anforderung des Masters reagieren können. Der Master sendet eine Anforderung an eine Slave-Adresse und nur die Slave-Adresse antwortet auf den Befehl.

### Modbus TCP-Kommunikation

Modbus TCP (Transmission Control Protocol) ist ein Standardprotokoll für die Kommunikation im Netzwerk und verwendet den PROFINET-Anschluss an der CPU für die TCP/IP-Kommunikation. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich.

Modbus TCP nutzt Open User Communications (OUC)-Verbindungen als Modbus-Kommunikationspfad. Neben der Verbindung zwischen STEP 7 und der CPU kann es mehrere Client/Server-Verbindungen geben. Gemischte Client- und Server-Verbindungen werden bis zur maximalen Anzahl der von der CPU zugelassenen Verbindungen (Seite 451) unterstützt.

Jede MB\_SERVER-Verbindung muss eine eindeutige Nummer für Instanz-DB und IP-Port verwenden. Je IP-Port wird nur eine Verbindung unterstützt. Für jede Verbindung muss MB\_SERVER (mit eindeutigem Instanz-DB und IP-Port) einzeln ausgeführt werden.

---

#### **Hinweis**

Modbus TCP funktioniert erst ab CPU Firmware Release V1.02 einwandfrei. Der Versuch, die Modbus-Anweisungen mit einer früheren Firmware-Version auszuführen, führt zu einem Fehler.

---

Ein Modbus TCP-Client (Master) muss die Client/Server-Verbindung über den Parameter DISCONNECT steuern. Die grundlegenden Aktionen eines Modbus-Clients werden im Folgenden gezeigt.

1. Verbindung zu einem Server (Slave) mit bestimmter IP-Adresse und IP-Portnummer aufbauen
2. Client-Übertragung von Modbus-Meldungen auslösen und Antworten vom Server empfangen
3. Gegebenenfalls die Verbindungsunterbrechung zwischen Client und Server auslösen, um die Verbindung mit einem anderen Server zu ermöglichen

#### **Modbus RTU-Anweisungen in Ihrem Programm**

- MB\_COMM\_LOAD: Eine Ausführung von MB\_COMM\_LOAD ist erforderlich, um PtP-Portparameter wie Baudrate, Parität und Flusskontrolle einzurichten. Nachdem der CPU-Port für das Modbus RTU-Protokoll konfiguriert ist, kann er nur von der Anweisung MB\_MASTER oder der Anweisung MB\_SLAVE verwendet werden.
- MB\_MASTER: Mit der Modbus-Master-Anweisung kann die CPU als Modbus-RTU-Mastergerät für die Kommunikation mit einem oder mehreren Modbus-Slavegeräten eingesetzt werden.
- MB\_SLAVE: Mit der Modbus-Slave-Anweisung kann die CPU als Modbus-RTU-Slavegerät für die Kommunikation mit einem Modbus-Mastergerät eingesetzt werden.

#### **Modbus TCP-Anweisungen in Ihrem Programm**

- MB\_CLIENT: Client/Server-TCP-Verbindung herstellen, Befehlsmeldung senden, Antwort empfangen und Trennen der Verbindung vom Server steuern
- MB\_SERVER: Bei Anforderung Verbindung zu einem Modbus-TCP-Client aufbauen, Modbus-Meldung empfangen und Antwort senden

## 12.5.2 Modbus TCP

### 12.5.2.1 MB\_CLIENT (Modbus TCP)

Tabelle 12- 49 Anweisung MB\_CLIENT

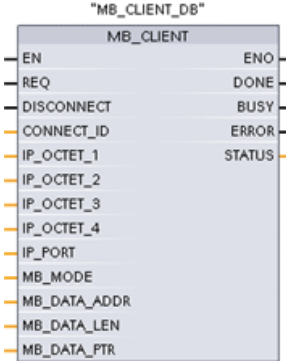
KOP/FUP	SCL	Beschreibung
	<pre>"MB_CLIENT_DB" (     REQ:= bool_in_,     DISCONNECT:= bool_in_,     CONNECT_ID:= uint_in_,     IP_OCTET_1:= byte_in_,     IP_OCTET_2:= byte_in_,     IP_OCTET_3:= byte_in_,     IP_OCTET_4:= byte_in_,     IP_PORT:= uint_in_,     MB_MODE:= usint_in_,     MB_DATA_ADDR:= udint_in_,     MB_DATA_LEN:= uint_in_,     DONE=&gt; bool_out_,     BUSY=&gt; bool_out_,     ERROR=&gt; bool_out_,     STATUS=&gt; word_out_,     MB_DATA_PTR:= variant inout );</pre>	<p>MB_CLIENT kommuniziert als Modbus TCP-Client über den PROFINET-Anschluss an der S7-1200 CPU. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich.</p> <p>MB_CLIENT kann eine Client-Server-Verbindung herstellen, eine Modbus-Funktionsanforderung senden, eine Antwort empfangen und das Trennen der Verbindung von einem Modbus TCP-Server steuern.</p>

Tabelle 12- 50 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	In	Bool FALSCH = Keine Modbus-Kommunikationsanforderung WAHR = Anforderung für die Kommunikation mit einem Modbus TCP-Server
DISCONNECT	IN	Bool Mit dem Parameter DISCONNECT kann Ihr Programm den Verbindungsaufbau und -abbau zu einem Modbus-Servergerät steuern. Ist DISCONNECT = 0 und keine Verbindung vorhanden, versucht MB_CLIENT eine Verbindung zur zugewiesenen IP-Adresse und Portnummer aufzubauen. Ist DISCONNECT = 1 und eine Verbindung vorhanden, wird versucht, die Verbindung zu trennen. Immer wenn dieser Eingang aktiviert ist, wird kein anderer Vorgang eingeleitet.
CONNECT_ID	IN	UInt Der Parameter CONNECT_ID muss jede Verbindung im PLC-Gerät eindeutig identifizieren. Jede eindeutige Instanz der Anweisung MB_CLIENT oder MB_SERVER muss einen eindeutigen Parameter CONNECT_ID enthalten.
IP_OCTET_1	IN	USInt IP-Adresse des Modbus TCP-Servers: Oktett 1 8-Bit-Anteil der 32-Bit-IP-Adresse (IPv4) des Modbus TCP-Servers, mit dem der Client über das Modbus TCP-Protokoll eine Verbindung herstellt und kommuniziert.
IP_OCTET_2	IN	USInt IP-Adresse des Modbus TCP-Servers: Oktett 2

Parameter und Datentyp		Datentyp	Beschreibung
IP_OCTET_3	IN	USInt	IP-Adresse des Modbus TCP-Servers: Oktett 3
IP_OCTET_4	IN	USInt	IP-Adresse des Modbus TCP-Servers: Oktett 4
IP_PORT	IN	UInt	Standardwert = 502: Die IP-Portnummer des Servers, mit dem der Client versucht, über das TCP/IP-Protokoll eine Verbindung herzustellen und anschließend zu kommunizieren.
MB_MODE	IN	USInt	Auswahl Modus: Weist die Art der Anforderung (Lesen, Schreiben oder Diagnose) zu. Weitere Informationen finden Sie in der Tabelle der Modbus-Funktionen unten.
MB_DATA_ADDR	IN	UDInt	Modbus-Anfangsadresse: Weist die Anfangsadresse der Daten zu, auf die MB_CLIENT zugreifen soll. Die gültigen Adressen finden Sie in der Tabelle der Modbus-Funktionen unten.
MB_DATA_LEN	IN	UInt	Modbus-Datenlänge: Weist die Anzahl der Bits oder Wörter zu, auf die diese Anforderung zugreifen soll. Die gültigen Längen finden Sie in der Tabelle der Modbus-Funktionen unten.
MB_DATA_PTR	IN_OUT	Variant	Pointer auf das Modbus-Datenregister: Das Register puffert Daten, die an einen Modbus-Server gesendet oder von dort empfangen werden. Der Pointer muss einen globalen Standard-DB oder eine Adresse im Speicherbereich der Merker zuweisen.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - Keine Anweisung MB_CLIENT in Bearbeitung</li> <li>• 1 - Anweisung MB_CLIENT in Bearbeitung</li> </ul>
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die Ausführung von MB_CLIENT mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

## Parameter REQ

FALSCH = Keine Modbus-Kommunikationsanforderung

WAHR = Anforderung für die Kommunikation mit einem Modbus TCP-Server

Wenn keine Instanz von MB\_CLIENT aktiv ist und der Parameter DISCONNECT = 0 ist, wird bei REQ = 1 eine neue Modbus-Anforderung gestartet. Ist die Verbindung noch nicht hergestellt, so wird eine neue Verbindung aufgebaut.

Wird dieselbe Instanz von MB\_CLIENT erneut mit DISCONNECT = 0 and REQ = 1 ausgeführt, bevor die aktuelle Anforderung abgearbeitet ist, werden keine nachfolgenden Modbus-Übertragungen durchgeführt. Sobald die aktuelle Anforderung jedoch abgearbeitet ist, kann eine neue Anforderung verarbeitet werden, sofern MB\_CLIENT mit REQ = 1 ausgeführt wird.

Wenn die aktuelle MB\_CLIENT-Kommunikationsanforderung abgearbeitet ist, ist das Bit DONE einen Zyklus lang WAHR. Das Bit DONE kann als Zeitfenster für die Sequenzierung mehrerer MB\_CLIENT-Anforderungen verwendet werden.

**Hinweis**

**Konsistenz der Eingangsdaten während der Verarbeitung von MB\_CLIENT**

Nachdem ein Modbus-Client eine Modbus-Operation initiiert, werden alle Eingangszustände intern gespeichert und dann bei jedem nachfolgenden Aufruf verglichen. Der Vergleich dient dazu, festzustellen, ob ein bestimmter Aufruf der ursprüngliche Auslöser der aktiven Client-Anforderung war. Mehrere Aufrufe von MB\_CLIENT können mit Hilfe eines gemeinsamen Instanz-DBs durchgeführt werden.

Deshalb ist es wichtig, dass die Eingänge während des Zeitraums, in dem eine MB\_CLIENT-Operation aktiv verarbeitet wird, nicht geändert werden. Wird diese Regel missachtet, kann MB\_CLIENT nicht feststellen, ob es die aktive Instanz ist.

**Über die Parameter MB\_MODE und MB\_DATA\_ADDR wählen Sie die Modbus-Kommunikationsfunktion aus.**

MB\_DATA\_ADDR weist die Modbus-Anfangsadresse der Daten zu, auf die zugegriffen werden soll. Die Anweisung MB\_CLIENT nutzt statt eines Funktionscodeeingangs einen Eingang MB\_MODE.

Die Kombination aus MB\_MODE- und MB\_DATA\_ADDR -Werten legt den Funktionscode fest, der in der eigentlichen Modbus-Meldung verwendet wird. Die folgende Tabelle zeigt die Entsprechung zwischen dem Parameter MB\_MODE, der Modbus-Funktion und dem Modbus-Adressbereich.

Tabelle 12- 51 Modbus-Funktionen

MB_MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	MB_DATA_ADDR
0	01	1 bis 2000	Ausgangsbits lesen: 1 bis 2000 Bits pro Anforderung	1 bis 9999
0	02	1 bis 2000	Eingangsbits lesen: 1 bis 2000 Bits pro Anforderung	10001 bis 19999
0	03	1 bis 125	Halteregister lesen: 1 bis 125 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
0	04	1 bis 125	Eingangswörter lesen: 1 bis 125 Wörter pro Anforderung	30001 bis 39999
1	05	1	Ein Ausgangsbit schreiben: Ein Bit pro Anforderung	1 bis 9999
1	06	1	Ein Halteregister schreiben: 1 Wort pro Anforderung	40001 bis 49999 oder 400001 bis 465535
1	15	2 nach 1968	Mehrere Ausgangsbits schreiben: 2 bis 1968 Bits pro Anforderung	1 bis 9999
1	16	2 bis 123	Mehrere Halteregister schreiben: 2 bis 123 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535



MB_MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	MB_DATA_ADDR
2	15	1 bis 1968	Ein oder mehrere Ausgangsbits schreiben: 1 bis 1968 Bits pro Anforderung	1 bis 9999
2	16	1 bis 123	Ein oder mehrere Haltereister schreiben: 1 bis 123 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
11	11	0	Statuswort und Ereigniszähler der Serverkommunikation lesen. Das Statuswort zeigt beschäftigt an (0 – nicht beschäftigt, 0xFFFF - beschäftigt). Der Ereigniszähler wird bei jeder erfolgreichen Abarbeitung einer Meldung inkrementiert.  Sowohl Parameter MB_DATA_ADDR als auch Parameter MB_DATA_LEN von MB_CLIENT wird bei dieser Funktion ignoriert.	
80	08	1	Serverzustand über Datendiagnosecode 0x0000 prüfen (Prüfschleifentest – Server gibt ein Echo der Anforderung zurück) 1 Wort pro Anforderung	
81	08	1	Server-Ereigniszähler über Datendiagnosecode 0x000A zurücksetzen 1 Wort pro Anforderung	
3 bis 10, 12 bis 79, 82 bis 255			Reserviert	

---

#### Hinweis

**MB\_DATA\_PTR weist einen Puffer zu, um aus einem Modbus TCP-Server gelesene bzw. in den Server geschriebene Daten zu speichern**

Der Datenpuffer kann sich in einem globalen Standard-DB oder in einer Adresse im Speicherbereich der Merker befinden.

Verwenden Sie für einen Puffer im Speicherbereich der Merker das Standardformat des Any-Pointer. Hierbei handelt es sich um das Format P#"Bitadresse" "Datentyp" "Länge". Ein Beispiel: P#M1000.0 WORD 500.

---

### Zuweisung eines Kommunikationspuffers durch MB\_DATA\_PTR

- Kommunikationsfunktionen von MB\_CLIENT:
  - 1-Bit-Daten aus Modbus-Serveradressen lesen und schreiben (00001 bis 09999)
  - 1-Bit-Daten aus Modbus-Serveradressen lesen (10001 bis 19999)
  - 16-Bit-Wortdaten aus Modbus-Serveradressen lesen (30001 bis 39999) und (40001 bis 49999)
  - 16-Bit-Wortdaten in Modbus-Serveradressen schreiben (40001 bis 49999)
- Daten in Wort- oder Bitgröße werden in den oder aus dem von MB\_DATA\_PTR zugewiesenen Puffer im DB oder Merkerbereich übertragen.
- Wenn von MB\_DATA\_PTR ein DB als Puffer zugewiesen ist, müssen Sie allen DB-Datenelementen Datentypen zuweisen.
  - Der 1-Bit-Datentyp Bool stellt eine Modbus-Bitadresse dar.
  - Datentypen mit einzelnen 16-Bit-Wörtern wie WORD, UInt und Int stellen eine Modbus-Wortadresse dar.
  - Datentypen mit 32-Bit-Doppelwörtern wie DWORD, DInt und Real stellen zwei Modbus-Wortadressen dar.
- Komplexe DB-Elemente können von MB\_DATA\_PTR zugewiesen werden, z.B.
  - Standard-Arrays
  - Benannte Strukturen, in denen jedes Element eindeutig ist.
  - Benannte komplexe Strukturen, in denen jedes Element einen eindeutigen Namen und einen 16- oder 32-Bit-Datentyp hat.
- Es ist nicht erforderlich, dass die Datenbereiche für MB\_DATA\_PTR in demselben globalen Datenbaustein (oder Speicherbereich der Merker) liegen. Sie können einen Datenbaustein für Modbus-Lesevorgänge, einen anderen Datenbaustein für Modbus-Schreibvorgänge oder einen Datenbaustein für jede MB\_CLIENT-Station anlegen.

### Mehrere Clientverbindungen

Ein Modbus TCP-Client unterstützt gleichzeitige Verbindungen bis zur maximalen für das PLC-Gerät zulässigen Anzahl von Open User Communications-Verbindungen. Die Gesamtzahl der Verbindungen für ein PLC-Gerät, einschließlich Modbus TCP-Clients und -Server, darf die maximale Anzahl der unterstützten Open User Communications-Verbindungen nicht überschreiten (Seite 451). Die Modbus TCP-Verbindungen können von Client- und/oder Serververbindungen gemeinsam genutzt werden.

Einzelne Clientverbindungen müssen die folgenden Regeln einhalten:

- Jede MB\_CLIENT-Verbindung muss einen unterschiedlichen Instanz-DB nutzen.
- Jede MB\_CLIENT-Verbindung muss eine eindeutige IP-Adresse eines Servers angeben.
- Jede MB\_CLIENT-Verbindung muss eine eindeutige Verbindungs-ID angeben.
- Eindeutige IP-Portnummern sind möglicherweise je nach Serverkonfiguration erforderlich.

Die Verbindungs-ID muss für jede einzelne Verbindung eindeutig sein. Das bedeutet, dass für die einzelnen Instanz-DBs nur jeweils eine einzelne, eindeutige Verbindungs-ID verwendet werden darf. Zusammengefasst heißt dies, dass Instanz-DB und Verbindungs-ID gepaart sind und für jede Verbindung eindeutig sein müssen.

Tabelle 12- 52 Für den Benutzer zugänglich statische Variablen des Instanz-Datenbausteins MB\_CLIENT

Variable	Datentyp	Voreinstellung	Beschreibung
Blocked_Proc_Timeout	Real	3.0	Zeitdauer (in Sekunden), die auf eine blockierte Modbus-Client-Instanz gewartet werden soll, bevor diese Instanz als AKTIV entfernt wird. Dies kann beispielsweise vorkommen, wenn eine Clientanforderung ausgegeben wurde und die Anwendung dann aufhört, die Clientfunktion auszuführen, bevor die Anforderung vollständig abgearbeitet wurde. Maximaler Grenzwert bei der S7-1200 sind 55 Sekunden.
MB_Unit_ID	Wort	255	Modbus-Geräteerkennung: Ein Modbus TCP-Server wird über seine IP-Adresse angesprochen. Deshalb wird der Parameter MB_UNIT_ID bei der Modbus TCP-Adressierung nicht verwendet.  Der Parameter MB_UNIT_ID entspricht dem Feld der Slaveadresse beim Modbus RTU-Protokoll. Wenn ein Modbus TCP-Server als Gateway zu einem Modbus RTU-Protokoll verwendet wird, kann das Slavegerät im seriellen Netzwerk über MB_UNIT_ID identifiziert werden. Der Parameter MB_UNIT_ID würde in diesem Fall die Anforderung an die richtige Modbus RTU-Slaveadresse weiterleiten.  Beachten Sie bitte, dass einige Modbus TCP-Geräte den Parameter MB_UNIT_ID möglicherweise für die Initialisierung innerhalb eines eingeschränkten Wertebereichs benötigen.
RCV_TIMEOUT	Real	2.0	Zeit in Sekunden, die MB_CLIENT auf die Antwort eines Servers auf eine Anforderung wartet.
Verbunden	Bool	0	Gibt an, ob die Verbindung zum zugewiesenen Server verbunden oder nicht verbunden ist: 1 = verbunden, 0 = nicht verbunden

Tabelle 12- 53 MB\_CLIENT-Protokollfehler

STATUS (W#16#)	Antwortcode an Modbus-Client (B#16#)	Modbus-Protokollfehler
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Zugriff außerhalb der Grenzen des Adressbereichs von MB_HOLD_REG
8384	03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)

Tabelle 12- 54 Bedingungs-codes der Ausführung von MB\_CLIENT <sup>1</sup>

STATUS (W#16#)	MB_CLIENT-Parameterfehler
7001	MB_CLIENT wartet auf die Antwort eines Modbus-Servers auf die Anforderung eines Verbindungsaufbaus oder -abbaus am zugewiesenen TCP-Port. Dies wird nur für die erste Ausführung eines Verbindungsaufbaus oder -abbaus gemeldet.
7002	MB_CLIENT wartet auf die Antwort eines Modbus-Servers auf die Anforderung eines Verbindungsaufbaus oder -abbaus am zugewiesenen TCP-Port. Dies wird für alle nachfolgenden Ausführungen gemeldet, während auf die Fertigstellung eines Verbindungsaufbaus oder -abbaus gewartet wird.
7003	Ein Verbindungsabbau wurde erfolgreich durchgeführt (nur einen PLC-Zyklus lang gültig).
80C8	Der Server hat nicht während der festgelegten Zeit reagiert. MB_CLIENT muss innerhalb der zugewiesenen Zeit über die ursprünglich gesendete Transaktions-ID eine Antwort erhalten oder es wird dieser Fehler ausgegeben. Prüfen Sie die Verbindung zum Modbus-Servergerät. Dieser Fehler wird erst gemeldet, nachdem konfigurierte Wiederholungen (sofern zutreffend) durchgeführt wurden.
8188	Moduswert ungültig
8189	Ungültiger Wert für die Datenadresse
818A	Ungültiger Wert für die Datenlänge
818B	Ungültiger Pointer auf den DATA_PTR-Bereich. Hierbei kann es sich um die Kombination aus MB_DATA_ADDRESS + MB_DATA_LEN handeln.
818C	Pointer auf einen optimierten DATA_PTR-Bereich (hierbei muss es sich um einen Standard-DB-Bereich oder um einen Bereich im Speicherbereich der Merker handeln)
8200	Der Port ist durch die Verarbeitung einer vorhandenen Modbus-Anforderung belegt.
8380	Der empfangene Modbus-Rahmen hat ein fehlerhaftes Format oder es wurden zu wenige Bytes empfangen.
8387	Der zugewiesene Parameter der Verbindungs-ID unterscheidet sich von der für vorherige Anforderungen verwendeten ID. Es darf in jedem Instanz-DB von MB_CLIENT nur eine einzige Verbindungs-ID verwendet werden. Dies ist auch ein interner Fehler, wenn die von einem Server empfangene Modbus TCP-Protokoll-ID nicht 0 ist.
8388	Ein Modbus-Server hat eine Datenmenge zurückgegeben, die sich von der angeforderten Menge unterscheidet. Dies gilt nur für die Modbus-Funktionen 15 und 16.

<sup>1</sup> Neben den oben aufgeführten MB\_CLIENT-Fehlern können auch von den zugrunde liegenden T-Bausteinanweisungen für die Kommunikation (TCON, TDISCON, TSEND und TRCV) Fehler gemeldet werden.

**Siehe auch**

TCON, TDISCON, TSEND UND TRCV (Seite 466)

## 12.5.2.2 MB\_SERVER (Modbus TCP)

Tabelle 12- 55 Anweisung MB\_SERVER

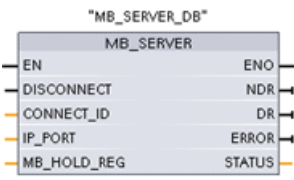
KOP/FUP	SCL	Beschreibung
	<pre>"MB_SERVER_DB" (   DISCONNECT:=_bool_in_,   CONNECT_ID:=_uint_in_,   IP_PORT:=_uint_in_,   NDR=&gt;_bool_out_,   DR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SERVER kommuniziert als Modbus TCP-Server über den PROFINET-Anschluss an der S7-1200 CPU. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich.</p> <p>MB_SERVER kann eine Anforderung für den Verbindungsaufbau mit einem Modbus TCP-Client annehmen, eine Modbus-Funktionsanforderung empfangen und eine Antwortmeldung senden.</p>

Tabelle 12- 56 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
DISCONNECT	IN	Bool MB_SERVER versucht, eine "passive" Verbindung mit einem Partnergerät aufzubauen. Das bedeutet, dass der Server passiv überwacht, ob es TCP-Verbindungsanforderungen von beliebigen, Anforderungen sendenden IP-Adressen gibt. Ist DISCONNECT = 0 und keine Verbindung vorhanden, kann eine passive Verbindung initiiert werden. Ist DISCONNECT = 1 und eine Verbindung vorhanden, wird ein Verbindungsabbau initiiert. Auf diese Weise kann Ihr Programm steuern, wann eine Verbindung angenommen wird. Immer wenn dieser Eingang aktiviert ist, wird kein anderer Vorgang eingeleitet.
CONNECT_ID	IN	UInt CONNECT_ID identifiziert jede Verbindung im PLC-Gerät eindeutig. Jede eindeutige Instanz der Anweisung MB_CLIENT oder MB_SERVER muss einen eindeutigen Parameter CONNECT_ID enthalten.
IP_PORT	IN	UInt Standardwert = 502: Die IP-Portnummer, die den Port identifiziert, der von einem Modbus-Client auf eine Verbindungsanforderung hin überwacht wird. Die folgenden TCP-Portnummern sind für eine passive MB_SERVER-Verbindung nicht zulässig: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 und 34964.
MB_HOLD_REG	IN_OUT	Variant Pointer auf das Modbus-Halteregister von MB_SERVER: Bei dem Halteregister muss es sich um einen globalen Standard-DB oder um eine Adresse im Speicherbereich der Merker handeln. Dieser Speicherbereich dient dazu, die Werte zu speichern, auf die ein Modbus-Client mit den Modbus-Registerfunktionen 3 (Lesen), 6 (Schreiben) und 16 (Schreiben) zugreifen kann.
NDR	OUT	Bool Neue Daten bereit: 0 = Keine neuen Daten, 1 = Gibt an, dass von einem Modbus-Client neue Daten geschrieben wurden
DR	OUT	Bool Daten lesen: 0 = Daten nicht gelesen, 1 = Gibt an, dass die Daten von einem Modbus-Client gelesen wurden

Parameter und Datentyp		Datentyp	Beschreibung
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die Ausführung von MB_SERVER mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

Bei MB\_SERVER können eingehende Modbus-Funktionscodes (1, 2, 4, 5 und 15) Bits und Wörter direkt in den Prozessabbildern der Eingänge und Ausgänge der S7-1200 CPU lesen und schreiben. Bei den Funktionscodes für die Datenübertragung (3, 6 und 16) muss der Parameter MB\_HOLD\_REG als Datentyp größer als ein Byte definiert sein. Die folgende Tabelle zeigt die Zuordnung der Modbus-Adressen zum Prozessabbild in der CPU.

Tabelle 12- 57 Zuordnung der Modbus-Adressen zum Prozessabbild

Modbus-Funktionen						S7-1200	
Codes	Funktion	Datenbereich	Adressbereich			Datenbereich	CPU-Adresse
01	Bits lesen	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
02	Bits lesen	Eingang	10001	An	18192	Prozessabbild der Eingänge	E0.0 bis E1023.7
04	Wörter lesen	Eingang	30001	An	30512	Prozessabbild der Eingänge	EW0 bis EW1022
05	Bit schreiben	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
15	Bits schreiben	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7

Funktionscodes eingehender Modbus-Meldungen (3, 6 und 16) lesen oder schreiben Wörter in einem Modbus-Halteregister, bei dem es sich um einen Adressbereich im Speicherbereich der Merker oder um einen Datenbaustein handeln kann. Der Typ des Haltesregisters wird vom Parameter MB\_HOLD\_REG angegeben.

**Hinweis**

**Zuweisung des Parameters MB\_HOLD\_REG**

Bei dem Modbus-Halteregister kann es sich um einen globalen Standard-DB oder um eine Adresse im Speicherbereich der Merker handeln.

Für ein Modbus-Halteregister im Speicherbereich der Merker verwenden Sie das Standardformat des Any-Pointer. Hierbei handelt es sich um das Format P#"Bitadresse" "Datentyp" "Länge". Ein Beispiel: P#M1000.0 WORD 500.

Die folgende Tabelle zeigt Beispiele für die Zuordnung von Modbus-Adressen zum Halteregister, das für die Modbus-Funktionscodes 03 (Wörter lesen), 06 (Wort schreiben) und 16 (Wörter schreiben) verwendet wird. Die tatsächliche obere Grenze der DB-Adressen wird vom maximalen Arbeitsspeicher und vom maximalen M-Speicher des jeweiligen CPU-Modells festgelegt.

Tabelle 12- 58 Beispiele für die Zuordnung von Modbus-Adressen zu Adressen im Speicher der CPU

Modbus-Adresse	Beispiele für Parameter von MB_HOLD_REG		
	P#M100.0 Word 5	P#DB10.DBx0.0 Word 5	"Rezept".Inhaltsstoff
40001	MW100	DB10.DBW0	"Rezept".Inhaltsstoff[1]
40002	MW102	DB10.DBW2	"Rezept".Inhaltsstoff[2]
40003	MW104	DB10.DBW4	"Rezept".Inhaltsstoff[3]
40004	MW106	DB10.DBW6	"Rezept".Inhaltsstoff[4]
40005	MW108	DB10.DBW8	"Rezept".Inhaltsstoff[5]

### Mehrere Serververbindungen

Es können mehrere Serververbindungen hergestellt werden. Dadurch ist es möglich, dass ein einziges PLC-Gerät gleichzeitig Verbindungen mit mehreren Modbus TCP-Clients herstellt.

Ein Modbus TCP-Server unterstützt gleichzeitige Verbindungen bis zur maximalen für das PLC-Gerät zulässigen Anzahl von Open User Communications-Verbindungen. Die Gesamtzahl der Verbindungen für ein PLC-Gerät, einschließlich Modbus TCP-Clients und -Server, darf die maximale Anzahl der unterstützten Open User Communications-Verbindungen nicht überschreiten (Seite 451). Die Modbus TCP-Verbindungen können von Client- und/oder Serververbindungen gemeinsam genutzt werden.

Einzelne Serververbindungen müssen die folgenden Regeln einhalten:

- Jede MB\_SERVER-Verbindung muss einen unterschiedlichen Instanz-DB nutzen.
- Jede MB\_SERVER-Verbindung muss mit einer eindeutigen IP-Portnummer hergestellt werden. Je Port wird nur eine Verbindung unterstützt.
- Jede MB\_SERVER-Verbindung muss eine eindeutige Verbindungs-ID verwenden.
- MB\_SERVER muss für jede Verbindung einzeln aufgerufen werden (mit dem entsprechenden Instanz-DB).

Die Verbindungs-ID muss für jede einzelne Verbindung eindeutig sein. Das bedeutet, dass nur mit jedem einzelnen Instanz-DB eine einzelne, eindeutige Verbindungs-ID verwendet werden darf. Zusammengefasst heißt dies, dass Instanz-DB und Verbindungs-ID gepaart sind und für jede Verbindung eindeutig sein müssen.

Tabelle 12- 59 Funktionscodes der Modbus-Diagnose

Modbus-Diagnosefunktionen von MB_SERVER		
Codes	Teilfunktion	Beschreibung
08	0x0000	Echotest Abfragedaten ausgeben: Die Anweisung MB_SERVER gibt einem Modbus-Client das Echo eines empfangenen Datenworts zurück.

Modbus-Diagnosefunktionen von MB_SERVER		
08	0x000A	Kommunikationsereigniszähler löschen: Die Anweisung MB_SERVER löscht den Kommunikationsereigniszähler, der für Modbus-Funktion 11 verwendet wird.
11		Kommunikationsereigniszähler abrufen: Die Anweisung MB_SERVER nutzt einen internen Kommunikationsereigniszähler, um die Anzahl erfolgreicher Modbus-Lese- und Modbus-Schreibenanforderungen, die an den Modbus-Server gesendet werden, zu erfassen. Der Zähler wird bei Funktion 8 oder Funktion 11 nicht hochgezählt. Auch bei Anforderungen, die zu einem Kommunikationsfehler führen, wird der Zähler nicht inkrementiert.  Die Broadcast-Funktion ist bei Modbus TCP nicht verfügbar, weil nur eine Client-Server-Verbindung zur Zeit vorhanden ist.

### Variablen von MB\_SERVER

Diese Tabelle zeigt die öffentlichen statischen Variablen im Instanz-Datenbaustein von MB\_SERVER, die in Ihrem Programm verwendet werden können.

Tabelle 12- 60 Öffentliche statische Variablen von MB\_SERVER

Variable	Datentyp	Standardwert	Beschreibung
HR_Start_Offset	Wort	0	Weist die Anfangsadresse des Modbus-Halteregisters zu
Request_Count	Wort	0	Die Anzahl aller von diesem Server empfangenen Anforderungen
Server_Message_Count	Wort	0	Die Anzahl der für diesen spezifischen Server empfangenen Anforderungen
Xmt_Rcv_Count	Wort	0	Die Anzahl der Sendungen oder Datenempfänge, bei denen ein Fehler aufgetreten ist. Wird auch inkrementiert, wenn eine Meldung empfangen wird, bei der es sich um eine ungültige Modbus-Meldung handelt.
Exception_Count	Wort	0	Modbus-spezifische Fehler, die eine zurückgegebene Ausnahme erfordern
Success_Count	Wort	0	Die Anzahl der für diesen spezifischen Server empfangenen Anforderungen ohne Protokollfehler
Verbunden	Bool	0	Gibt an, ob die Verbindung zum zugewiesenen Client verbunden oder nicht verbunden ist: 1 = verbunden, 0 = nicht verbunden

Ihr Programm kann in HR\_Start\_Offset Werte schreiben und die Modbus-Server-Operationen steuern. Die anderen Variablen können gelesen werden, um den Modbus-Zustand zu überwachen.

### HR\_Start\_Offset

Die Adressen des Modbus-Halteregisters beginnen bei 40001. Diese Adressen entsprechen der Anfangsadresse des Halteregisters im Zielsystemspeicher. Sie können jedoch die Variable HR\_Start\_Offset konfigurieren, um eine andere Anfangsadresse als 40001 für das Modbus-Halteregister zu konfigurieren.

Sie können z. B. ein Halteregister mit Beginn an MW100 und einer Länge von 100 Wörtern konfigurieren. Mit einem Versatz von 20 geben Sie eine Anfangsadresse des Halteregisters von 40021 statt 40001 an. Jede Adresse unter 40021 und über 40119 führt zu einem Adressierungsfehler.

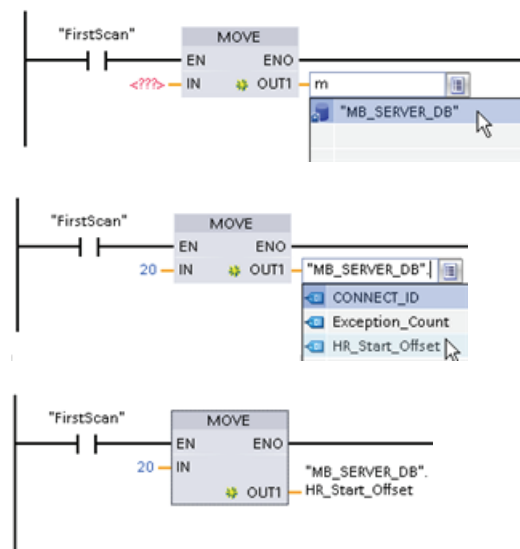


Tabelle 12- 61 Beispiel für die Adressierung des Modbus-Halteregisters

HR_Start_Offset	Adresse	Minimum	Maximum
0	Modbus-Adresse (Wort)	40001	40099
	S7-1200 Adresse	MW100	MW298
20	Modbus-Adresse (Wort)	40021	40119
	S7-1200 Adresse	MW100	MW298

HR\_Start\_Offset ist ein Wortwert, der die Anfangsadresse des Modbus-Halteregisters angibt und im Instanz-Datenbaustein MB\_SERVER gespeichert ist. Sie können diese öffentliche statische Variable über die Parameter-Klappliste auswählen, nachdem Sie MB\_SERVER in Ihr Programm eingefügt haben.

Wenn Sie beispielsweise MB\_SERVER in ein KOP-Netzwerk eingefügt haben, können Sie in ein vorheriges Netzwerk gehen und den Wert HR\_Start\_Offset zuweisen. Der Wert muss vor der Ausführung von MB\_SERVER zugewiesen werden.



Eingabe einer Variable für den Modbus-Server

über den Namen des Standard-DB:

1. Positionieren Sie den Cursor im Parameterfeld und geben Sie das Zeichen m ein.
2. Wählen Sie in der Klappliste der DB-Namen "MB\_SERVER\_DB" aus.
3. Wählen Sie in der Klappliste der DB-Variablen "MB\_SERVER\_DB.HR\_Start\_Offset" aus.

Tabelle 12- 62 Bedingungscode der Ausführung von MB\_SERVER 1

STATUS (W#16#)	Antwortcode an Modbus-Server (B#16#)	Modbus-Protokollfehler
7001		MB_SERVER wartet darauf, dass ein Modbus-Client die Verbindung zum zugewiesenen TCP-Port herstellt. Dieser Code wird bei der ersten Ausführung eines Verbindungsaufbaus oder -abbaus gemeldet.
7002		MB_SERVER wartet darauf, dass ein Modbus-Client die Verbindung zum zugewiesenen TCP-Port herstellt. Dieser Code wird für alle nachfolgenden Ausführungen gemeldet, während auf die Fertigstellung eines Verbindungsaufbaus oder -abbaus gewartet wird.

STATUS (W#16#)	Antwortcode an Modbus-Server (B#16#)	Modbus-Protokollfehler
7003		Ein Verbindungsabbau wurde erfolgreich durchgeführt (nur einen PLC-Zyklus lang gültig).
8187		Ungültiger Pointer auf MB_HOLD_REG: Bereich ist zu klein
818C		Pointer auf einen optimierten MB_HOLD_REG-Bereich (hierbei muss es sich um einen Standard-DB-Bereich oder um einen Bereich im Speicherbereich der Merker handeln) oder Timeout für gesperrten Prozess überschreitet den Grenzwert von 55 Sekunden. (spezifisch für die S7-1200)
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Zugriff außerhalb der Grenzen des Adressbereichs von MB_HOLD_REG
8384	03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)

<sup>1</sup> Neben den oben aufgeführten MB\_SERVER-Fehlern können Fehler auch von den zugrunde liegenden T-Bausteinanweisungen für die Kommunikation (TCON, TDISCON, TSEND und TRCV) gemeldet werden.

**Siehe auch**

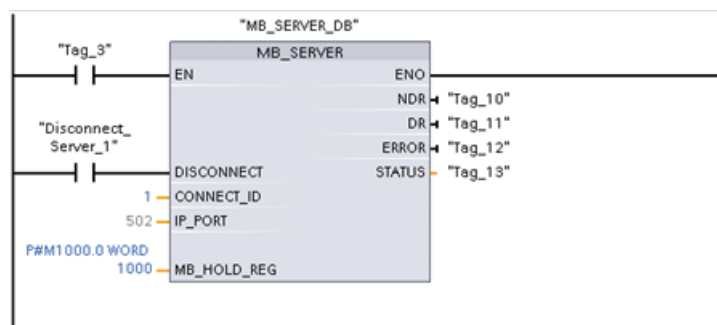
TCON, TDISCON, TSEND UND TRCV (Seite 466)

**12.5.2.3 Beispiel MB\_SERVER: Mehrere TCP-Verbindungen**

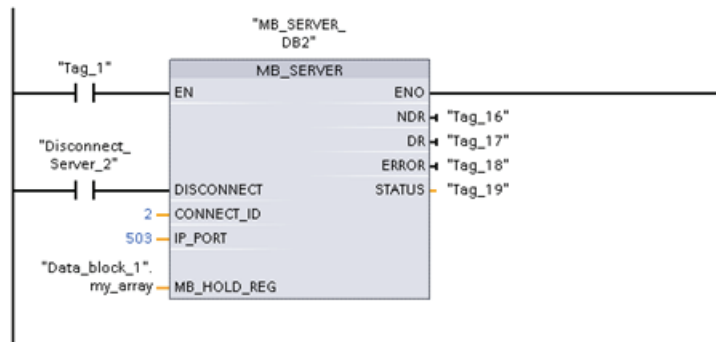
Sie können mehrere Verbindungen zum Modbus TCP-Server haben. Hierfür muss MB\_SERVER für jede Verbindung unabhängig ausgeführt werden. Jede Verbindung muss einen unabhängigen Instanz-DB, eine Verbindungs-ID und einen IP-Port verwenden. Bei der S7-1200 ist nur je eine Verbindung pro IP-Port zulässig.

Um optimales Betriebsverhalten zu erzielen, muss MB\_SERVER in jedem Zyklus für jede Verbindung ausgeführt werden.

**Netzwerk 1:** Verbindung Nr. 1 mit unabhängigem IP\_PORT, Verbindungs-ID und Instanz-DB



**Netzwerk 2:** Verbindung Nr. 2 mit unabhängigem IP\_PORT, Verbindungs-ID und Instanz-DB



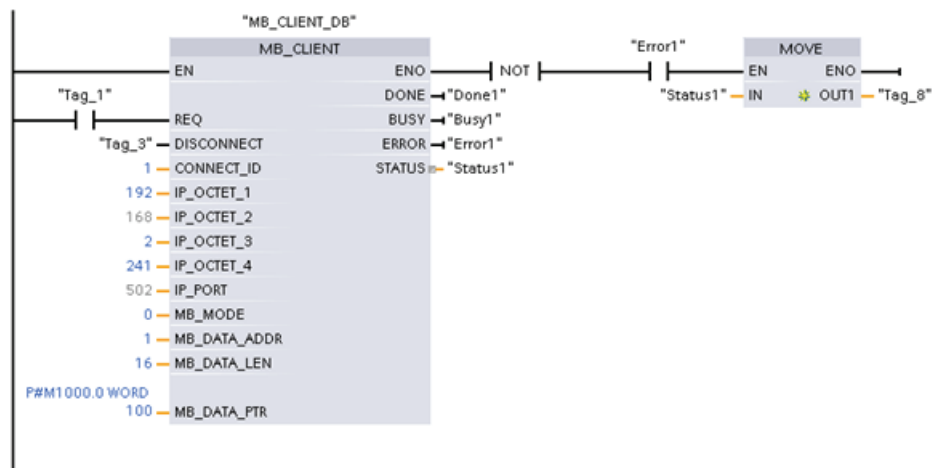
**12.5.2.4 Beispiel 1 für MB\_CLIENT: Mehrere Anforderungen mit gemeinsamer TCP-Verbindung**

Mehrere Modbus-Clientanforderungen können über die gleiche Verbindung gesendet werden. Hierfür verwenden Sie den gleichen Instanz-DB, die gleiche Verbindungs-ID und Portnummer.

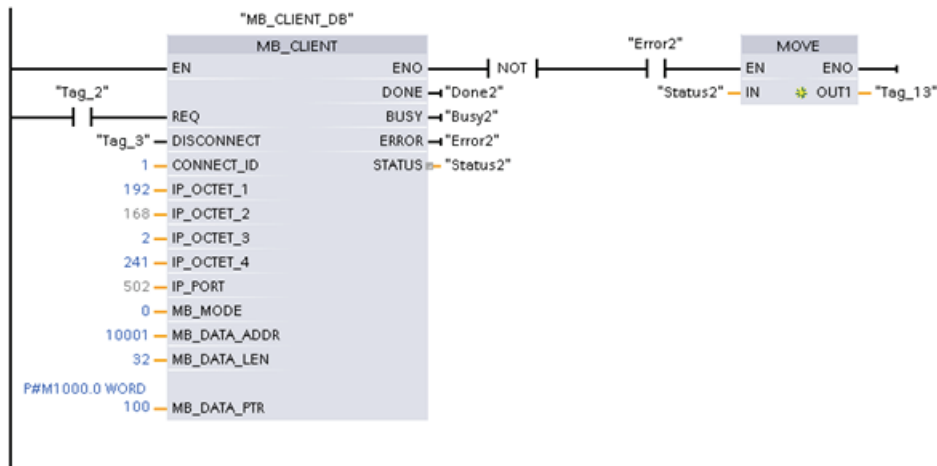
Es kann jeweils nur 1 Client aktiv sein. Nachdem ein Client seine Ausführung beendet hat, beginnt der nächste Client mit der Ausführung. Ihr Programm ist für die Reihenfolge der Ausführung verantwortlich.

Im Beispiel schreiben beide Clients in denselben Speicherbereich. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

**Netzwerk 1:** Modbus-Funktion 1 - 16 Bits im Prozessabbild der Ausgänge lesen



**Netzwerk 2:** Modbus-Funktion 2 - 32 Bits im Prozessabbild der Eingänge lesen



**12.5.2.5 Beispiel 2 für MB\_CLIENT: Mehrere Anforderungen mit unterschiedlichen TCP-Verbindungen**

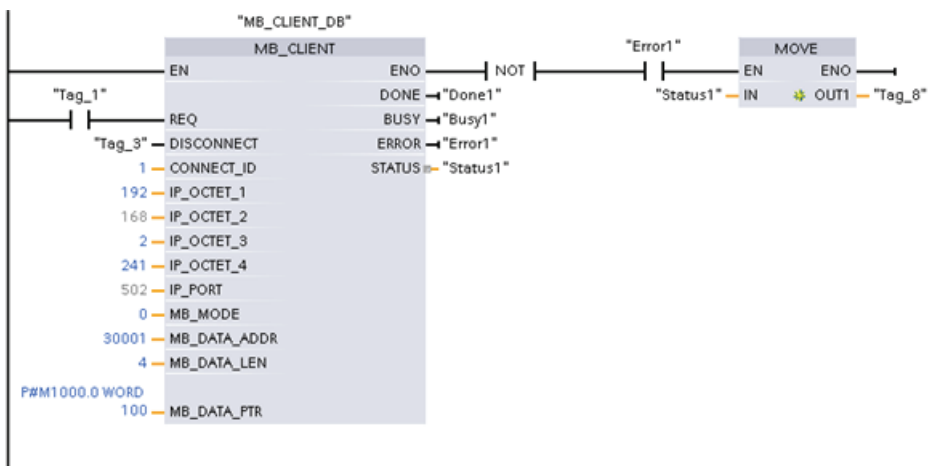
Mehrere Modbus-Clientanforderungen können über verschiedene Verbindungen gesendet werden. Hierfür müssen unterschiedliche Instanz-DBs, IP-Adressen und Verbindungs-IDs verwendet werden.

Die Portnummer muss unterschiedlich sein, wenn die Verbindungen mit demselben Modbus-Server aufgebaut werden. Bei Verbindungen zu unterschiedlichen Servern gibt es keine Einschränkungen hinsichtlich der Portnummer.

Im Beispiel schreiben beide Clients in denselben Speicherbereich. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

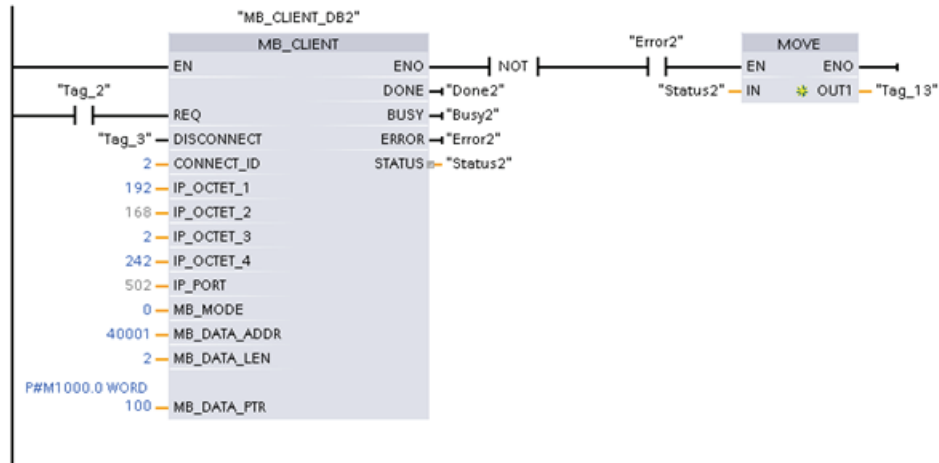
**Netzwerk 1:**

Modbus-Funktion 4 - Eingangswörter lesen (im Speicher der S7-1200)



### Netzwerk 2:

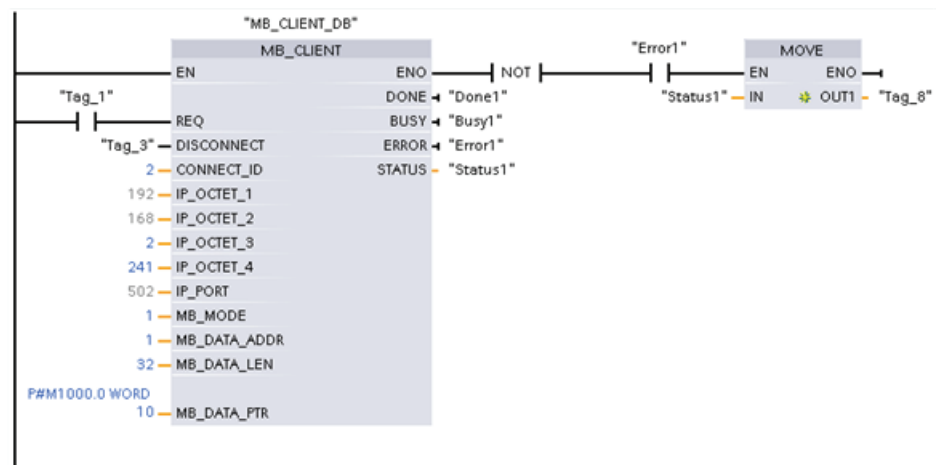
Modbus-Funktion 3 - Wörter im Haltereister lesen (im Speicher der S7-1200)



### 12.5.2.6 Beispiel 3 für MB\_CLIENT: Schreibenanforderung für das Prozessabbild der Ausgänge

Dieses Beispiel zeigt die Anforderung eines Modbus-Clients zum Schreiben in das Prozessabbild der Ausgänge der S7-1200.

**Netzwerk 1:** Modbus-Funktion 15 - Bits in das Prozessabbild der Ausgänge der S7-1200 schreiben

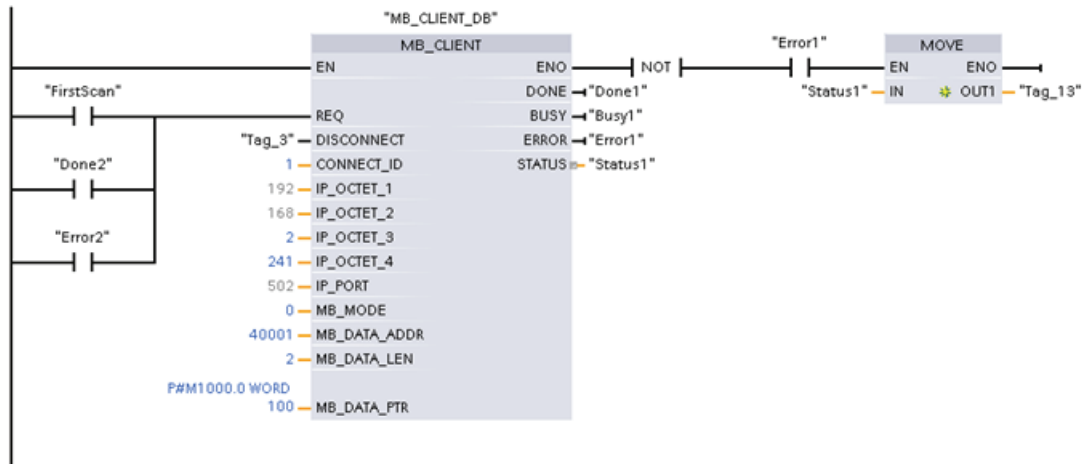


### 12.5.2.7 Beispiel 4 für MB\_CLIENT: Mehrere Anforderungen koordinieren

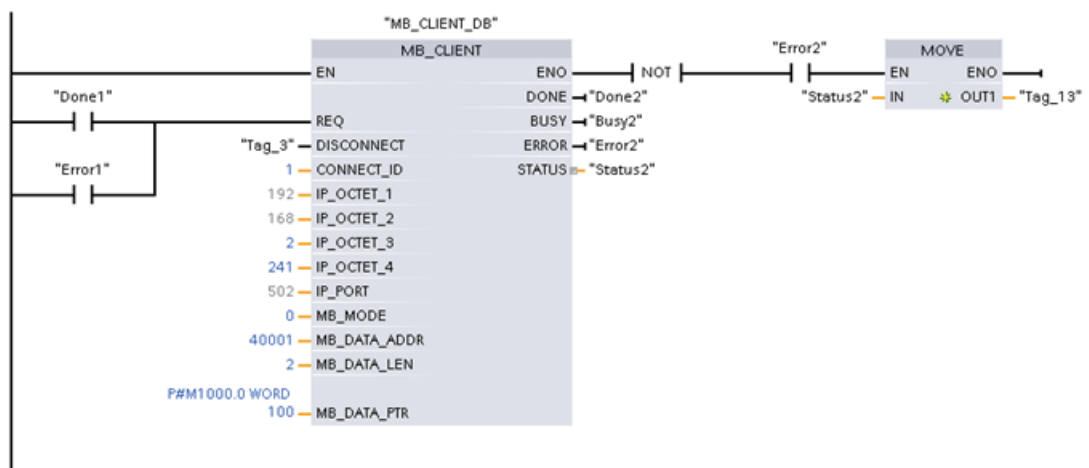
Sie müssen sicherstellen, dass die Ausführung jeder einzelnen Modbus TCP-Anforderung abgeschlossen wird. Diese Koordination hat durch Ihr Programm zu erfolgen. Das folgende Beispiel zeigt, wie die Ausführung über die Ausgänge der ersten und der zweiten Clientanforderung koordiniert werden kann.

Im Beispiel schreiben beide Clients in denselben Speicherbereich. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

**Netzwerk 1: Modbus-Funktion 3 - Wörter im Halteregister lesen**



**Netzwerk 2: Modbus-Funktion 3 - Wörter im Halteregister lesen**



**12.5.3 Modbus RTU**

Es gibt zwei Versionen der Modbus RTU-Anweisungen in STEP 7:

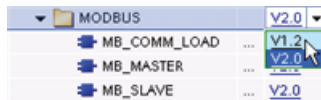
- Version 1 war zunächst in STEP 7 Basic V10.5 verfügbar.
- Version 2 ist in STEP 7 Basic/Professional V11 verfügbar. Die Version 2 bietet die zusätzlichen Parameter REQ und DONE bei MB\_COMM\_LOAD. Außerdem gestattet der Parameter MB\_ADDR bei MB\_MASTER und MB\_SLAVE nun einen UInt-Wert für die erweiterte Adressierung.

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Verwenden Sie niemals Anweisungen der Version 1.x und 2.y in einem CPU-Programm. Die Modbus-Anweisungen in Ihrem Programm müssen dieselbe Versionsnummer haben (1.x, 2.y oder V.z). Die einzelnen Anweisungen in einer Versionsgruppe können dann verschiedene Unterversionen aufweisen (1.x).



Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.



Um die Version einer Modbus-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine Modbus-Anweisung über das Anweisungsverzeichnis in Ihr Programm einfügen, wird in der Projektnavigation eine neue FB-Instanz angelegt. Die neue FB-Instanz finden Sie in der Projektnavigation unter PLC\_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer Modbus-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmierer angezeigten Box. Wählen Sie in der Projektnavigation die Instanz eines Modbus-FBs aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der Modbus-Anweisung anzuzeigen.

### 12.5.3.1 MB\_COMM\_LOAD

Tabelle 12- 63 Anweisung MB\_COMM\_LOAD

KOP/FUP	SCL	Beschreibung
	<pre>"MB_COMM_LOAD_DB" (     REQ:=_bool_in_,     PORT:=_uint_in_,     BAUD:=_udint_in_,     PARITY:=_uint_in_,     FLOW_CTRL:=_uint_in_,     RTS_ON_DLY:=_uint_in_,     RTS_OFF_DLY:=_uint_in_,     RESP_TO:=_uint_in_,     DONE=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_,     MB_DB:=_fbtref_inout_);</pre>	<p>Die Anweisung MB_COMM_LOAD konfiguriert einen PTP-Port für die Kommunikation über das Modbus-RTU-Protokoll. Hardwareoptionen des Modbus-Ports: Einbau von max. drei CMs (RS485 oder RS232), plus eines CB (R4845). Wenn Sie die Anweisung MB_COMM_LOAD in Ihr Programm einfügen, wird automatisch ein Instanz-Datenbaustein zugewiesen.</p>

Tabelle 12- 64 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Nur Version 2.0)
PORT	IN	Port	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen.
BAUD	IN	UDInt	Auswahl der Baudrate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Alle anderen Werte sind ungültig.
PARITY	IN	UInt	Auswahl der Parität: <ul style="list-style-type: none"> <li>• 0 – Keine</li> <li>• 1 – Ungerade</li> <li>• 2 – Gerade</li> </ul>
FLOW_CTRL	IN	UInt	Auswahl der Flusskontrolle: <ul style="list-style-type: none"> <li>• 0 – (Standard) Keine Flusskontrolle</li> <li>• 1 – Hardware-Flusskontrolle mit RTS immer EIN (gilt nicht bei RS485-Ports)</li> <li>• 2 – Hardware-Flusskontrolle mit RTS geschaltet</li> </ul>
RTS_ON_DLY	IN	UInt	Auswahl RTS-Einschaltverzögerung: <ul style="list-style-type: none"> <li>• 0 – (Standard) Keine Verzögerung von "RTS aktiv", bis das erste Zeichnung der Meldung gesendet wird.</li> <li>• 1 bis 65535 – Verzögerung in Millisekunden von "RTS aktiv", bis das erste Zeichen der Meldung gesendet wird (gilt nicht bei RS485-Ports). RTS-Verzögerungen müssen unabhängig von der Auswahl FLOW_CTRL angewendet werden.</li> </ul>
RTS_OFF_DLY	IN	UInt	Auswahl RTS-Ausschaltverzögerung: <ul style="list-style-type: none"> <li>• 0 – (Standard) Keine Verzögerung nach der Übertragung des letzten Zeichens bis "RTS inaktiv"</li> <li>• 1 bis 65535 – Verzögerung in Millisekunden von der Übertragung des letzten Zeichens bis "RTS inaktiv" (gilt nicht bei RS485-Ports). RTS-Verzögerungen müssen unabhängig von der Auswahl FLOW_CTRL angewendet werden.</li> </ul>
RESP_TO	IN	UInt	Antwort-Timeout: Zeit in Millisekunden, die MB_MASTER auf eine Antwort vom Slave wartet. Wenn der Slave während dieses Zeitraums nicht antwortet, wiederholt MB_MASTER die Anforderung oder beendet die Anforderung mit einem Fehler, wenn die angegebene Anzahl von Wiederholungen gesendet wurde. 5 ms bis 65535 ms (Standardwert = 1000 ms).
MB_DB	IN	Variant	Ein Verweis auf den Instanz-Datenbaustein der Anweisungen MB_MASTER oder MB_SLAVE. Nachdem Sie MB_SLAVE oder MB_MASTER in Ihr Programm eingefügt haben, ist die DB-Kennung in der Parameter-Klappliste am Anschluss der MB_DB-Box verfügbar.



Parameter und Datentyp		Datentyp	Beschreibung
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Nur Version 2.0)
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

MB\_COMM\_LOAD wird ausgeführt, um einen Port für das Modbus-RTU-Protokoll zu konfigurieren. Nachdem ein Port für das Modbus-RTU-Protokoll konfiguriert ist, kann er nur von den Anweisungen MB\_MASTER oder MB\_SLAVE verwendet werden.

Für die Konfiguration jedes Kommunikationsports, der für die Modbus-Kommunikation eingesetzt wird, muss eine Ausführung von MB\_COMM\_LOAD genutzt werden. Sie müssen jedem Port, den Sie nutzen, einen eindeutigen Instanz-DB MB\_COMM\_LOAD zuweisen. Sie können bis zu drei Kommunikationsmodule (RS232 oder RS485) und ein Kommunikationsboard (RS485) in die CPU einbauen. Sie können MB\_COMM\_LOAD aus einem Anlauf-OB aufrufen und einmal ausführen, oder Sie können den Aufruf für die einmalige Ausführung mit Hilfe des Systemmerkers für den ersten Zyklus (Seite 92) initiieren. Führen Sie MB\_COMM\_LOAD nur dann erneut aus, wenn Kommunikationsparameter wie Baudrate oder Parität geändert werden müssen.

Wenn Sie MB\_MASTER oder MB\_SLAVE in Ihr Programm einfügen, wird der Anweisung ein Instanz-Datenbaustein zugewiesen. Dieser Instanz-Datenbaustein wird referenziert, wenn Sie in der Anweisung MB\_COMM\_LOAD den Parameter MB\_DB angeben.

### MB\_COMM\_LOAD Datenbausteinvariablen

Die folgende Tabelle zeigt die öffentlichen statischen Variablen im Instanz-DB von MB\_COMM\_LOAD, die in Ihrem Programm verwendet werden können.

Tabelle 12- 65 Statische Variablen im Instanz-DB

Variable	Datentyp	Beschreibung
ICHAR_GAP	Word	Verzögerung für den Zeichenabstand zwischen den Zeichen. Dieser Parameter wird in Millisekunden angegeben und dient dazu, den erwarteten Zeitraum zwischen den empfangenen Zeichen zu erhöhen. Die entsprechende Anzahl Bitzeiten für diesen Parameter wird zum Modbus-Standardwert von 35 Bitzeiten (3,5 Zeichenzeiten) addiert.
RETRIES	Wort	Die Anzahl der wiederholten Versuche, die der Master durchführt, bevor der Fehlercode 0x80C8 für "Keine Antwort" zurückgegeben wird.

Tabelle 12- 66 Bedingungscode der Ausführung von MB\_COMM\_LOAD <sup>1</sup>

STATUS (W#16#)	Beschreibung
0000	Kein Fehler
8180	Ungültiger Wert der Port-ID (falscher Port/falsche Hardwarekennung des Kommunikationsmoduls)
8181	Ungültiger Wert für die Baudrate.
8182	Ungültiger Wert für die Parität.

STATUS (W#16#)	Beschreibung
8183	Ungültiger Wert für die Flusskontrolle.
8184	Ungültiger Wert für die Timeout-Zeit der Antwort (Antwort-Timeout ist kleiner als der Mindestwert von 5 ms)
8185	Der Parameter MB_DB ist kein Instanz-Datenbaustein einer Anweisung MB_MASTER oder MB_SLAVE.

<sup>1</sup> Neben den oben aufgeführten Fehlern der Anweisung MB\_COMM\_LOAD können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

**Siehe auch**

Punkt-zu-Punkt-Operationen (Seite 600)

**12.5.3.2 MB\_MASTER**

Tabelle 12- 67 Anweisung MB\_MASTER

KOP/FUP	SCL	Beschreibung
	<pre>"MB_MASTER_DB" (   REQ:= _bool_in_,   MB_ADDR:= _uint_in_,   MODE:= _usint_in_,   DATA_ADDR:= _udint_in_,   DATA_LEN:= _uint_in_,   DONE=&gt; _bool_out_,   BUSY=&gt; _bool_out_,   ERROR=&gt; _bool_out_,   STATUS=&gt; _word_out_,   DATA_PTR:= variant inout );</pre>	<p>Die Anweisung MB_MASTER kommuniziert als Modbus-Master über einen Port, der von einer vorherigen Ausführung der Anweisung MB_COMM_LOAD konfiguriert wurde. Wenn Sie die Anweisung MB_MASTER in Ihr Programm einfügen, wird automatisch ein Instanz-Datenbaustein zugewiesen. Dieser Instanz-Datenbaustein MB_MASTER wird verwendet, wenn Sie in der Anweisung MB_COMM_LOAD den Parameter MB_DB angeben.</p>

Tabelle 12- 68 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool 0 = Keine Anforderung 1 = Anforderung, Daten an den Modbus-Slave zu senden
MB_ADDR	IN	V1.0: USInt V2.0: UInt Modbus-RTU-Stationsadresse: Standard-Adressierungsbereich (1 bis 247) Erweiterter Adressierungsbereich (1 bis 65535) Der Wert 0 ist für den Broadcast einer Meldung an alle Modbus-Slaves reserviert. Für den Broadcast werden nur die Modbus-Funktionscodes 05, 06, 15 und 16 unterstützt.
MODE	IN	USInt Auswahl Modus: Gibt die Art der Anforderung an (Lesen, Schreiben oder Diagnose). Weitere Informationen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_ADDR	IN	UDInt Anfangsadresse im Slave: Gibt die Anfangsadresse der Daten an, auf die im Modbus-Slave zugegriffen werden soll. Die gültigen Adressen finden Sie in der Tabelle der Modbus-Funktionen unten.

Parameter und Datentyp		Datentyp	Beschreibung
DATA_LEN	IN	UInt	Datenlänge: Gibt die Anzahl der Bits oder Wörter an, auf die diese Anforderung zugreifen soll. Die gültigen Längen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_PTR	IN	Variant	Daten-Pointer: Zeigt auf die M- oder DB-Adresse (Standard-DB) für die zu schreibenden oder zu lesenden Daten.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 – Keine Anweisung MB_MASTER in Bearbeitung</li> <li>• 1 – Anweisung MB_MASTER in Bearbeitung</li> </ul>
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

### Regeln für die Kommunikation des Modbus-Masters

- MB\_COMM\_LOAD muss ausgeführt werden, um einen Port zu konfigurieren, damit die Anweisung MB\_MASTER mit diesem Port kommunizieren kann.
- Wenn ein Port zum Initiieren von Modbus-Masteranforderungen verwendet werden soll, darf dieser Port nicht von MB\_SLAVE verwendet werden. Eine oder mehrere Instanzen von MB\_MASTER können mit diesem Port verwendet werden. Doch alle Ausführungen von MB\_MASTER müssen den gleichen Instanz-DB MB\_MASTER für den Port verwenden.
- Die Modbus-Anweisungen nutzen keine Kommunikations-Alarmereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss die Anweisung MB\_MASTER auf abgeschlossene Sende- und Empfangsvorgänge abfragen.

- Es wird empfohlen, alle Ausführungen von MB\_MASTER für einen bestimmten Port aus einem Programmzyklus-OB aufzurufen. Modbus-Master-Anweisungen können nur in jeweils einem Programmzyklus oder in jeweils einer zyklischen/zeitverzögerten Ausführungsschicht ausgeführt werden. Sie dürfen nicht in beiden Prioritätsschichten der Ausführung bearbeitet werden. Die Vorrangunterbrechung einer Modbus-Master-Anweisung durch eine andere Modbus-Master-Anweisung in einer Ausführungsschicht mit höherer Priorität führt zu nicht ordnungsgemäßigem Betrieb. Die Modbus-Master-Anweisungen dürfen nicht in Anlauf-, Diagnose- oder Zeitfehlerausführungsschichten bearbeitet werden.
- Nachdem eine Masteranweisung eine Übertragung ausgelöst hat, muss diese Instanz mit aktiviertem Eingang EN fortlaufend ausgeführt werden, bis der Zustand DONE = 1 oder ERROR = 1 ausgegeben wird. Eine bestimmte MB\_MASTER-Instanz wird als aktiv betrachtet, bis eines dieser beiden Ereignisse auftritt. Während die ursprüngliche Instanz aktiv ist, führt jeder Aufruf einer weiteren Instanz mit aktiviertem Eingang REQ zu einem Fehler. Wenn die fortlaufende Ausführung der ursprünglichen Instanz stoppt, bleibt der Anforderungszustand für die von der statischen Variable Blocked\_Proc\_Timeout angegebene Zeitdauer aktiv. Nach Ablauf dieser Zeitdauer wird die nächste aufgerufene Master-Anweisung mit aktiviertem Eingang REQ zur aktiven Instanz. Dies verhindert, dass eine einzige Modbus-Master-Anweisung ein Monopol inne hat oder den Zugriff auf den Port sperrt. Wenn die ursprüngliche aktive Instanz nicht während des von der statischen Variablen "Blocked\_Proc\_Timeout" angegebenen Zeitraums aktiviert wird, löscht die nächste Ausführung durch diese Instanz (ohne gesetzten Eingang REQ) den aktiven Zustand. Falls (REQ gesetzt ist), dann löst diese Ausführung eine neue Master-Anforderung aus, als wenn keine andere Instanz aktiv wäre.

### Parameter REQ

0 = Keine Anforderung; 1 = Anforderung, Daten an den Modbus-Slave zu senden

Sie können diesen Eingang über einen pegel- oder flankengesteuerten Kontakt steuern. Immer wenn dieser Eingang aktiviert wird, wird eine Zustandsmaschine gestartet, um sicherzustellen, dass eine andere Anweisung MB\_MASTER, die denselben Instanz-DB verwendet, erst dann eine Anforderung abgeben kann, wenn die aktuelle Anforderung abgearbeitet ist. Alle anderen Eingangszustände werden erfasst und intern für die aktuelle Anforderung gespeichert, bis die Antwort empfangen oder ein Fehler erkannt wird.

Wird dieselbe Instanz von MB\_MASTER erneut mit dem REQ-Eingang = 1 ausgeführt, bevor die aktuelle Anforderung abgearbeitet ist, werden keine nachfolgenden Übertragungen durchgeführt. Wenn die Anforderung jedoch abgearbeitet ist, wird zu dem Zeitpunkt, zu dem MB\_MASTER erneut mit REQ-Eingang = 1 ausgeführt wird, eine neue Anforderung abgegeben.

### Über die Parameter DATA\_ADDR und MODE wählen Sie den Modbus-Funktionstyp aus.

DATA\_ADDR (Modbus-Anfangsadresse im Slave): Gibt die Anfangsadresse der Daten an, auf die im Modbus-Slave zugegriffen werden soll.

Die Anweisung MB\_MASTER nutzt statt eines Funktionscodeeingangs einen Eingang MODE. Die Kombination aus MODE und Modbus-Adresse legt den Funktionscode fest, der in der eigentlichen Modbus-Meldung verwendet wird. Die folgende Tabelle zeigt die Entsprechung zwischen dem Parameter MODE, dem Modbus-Funktionscode und dem Modbus-Adressbereich.

Tabelle 12- 69 Modbus-Funktionen

MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	Modbus-Adresse
0	01	1 bis 2000 1 bis 1992 <sup>1</sup>	Ausgangsbits lesen: 1 bis (1992 oder 2000) Bits pro Anforderung	1 bis 9999
0	02	1 bis 2000 1 bis 1992 <sup>1</sup>	Eingangsbits lesen: 1 bis (1992 oder 2000) Bits pro Anforderung	10001 bis 19999
0	03	1 bis 125 1 bis 124 <sup>1</sup>	Halteregister lesen: 1 bis (124 oder 125) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
0	04	1 bis 125 1 bis 124 <sup>1</sup>	Eingangswörter lesen: 1 bis (124 oder 125) Wörter pro Anforderung	30001 bis 39999
1	05	1	Ein Ausgangsbit schreiben: Ein Bit pro Anforderung	1 bis 9999
1	06	1	Ein Halteregister schreiben: 1 Wort pro Anforderung	40001 bis 49999 oder 400001 bis 465535
1	15	2 bis 1968 2 bis 1960 <sup>1</sup>	Mehrere Ausgangsbits schreiben: 2 bis (1960 oder 1968) Bits pro Anforderung	1 bis 9999
1	16	2 bis 123 2 bis 122 <sup>1</sup>	Mehrere Halteregister schreiben: 2 bis (122 oder 123) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
2	15	1 bis 1968 2 bis 1960 <sup>1</sup>	Ein oder mehrere Ausgangsbits schreiben: 1 bis (1960 oder 1968) Bits pro Anforderung	1 bis 9999
2	16	1 bis 123 1 bis 122 <sup>1</sup>	Ein oder mehrere Halteregister schreiben: 1 bis (122 oder 123) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
11	11	0	Statuswort und Ereigniszähler der Slave-Kommunikation lesen. Das Statuswort zeigt beschäftigt an (0 – nicht beschäftigt, 0xFFFF - beschäftigt). Der Ereigniszähler wird bei jeder erfolgreichen Abarbeitung einer Meldung inkrementiert.  Beide Operanden DATA_ADDR und DATA_LEN von MB_MASTER werden bei dieser Funktion ignoriert.	
80	08	1	Slave-Zustand über Daten-Diagnosecode 0x0000 prüfen (Prüfschleifentest – Slave gibt ein Echo der Anforderung zurück)  1 Wort pro Anforderung	
81	08	1	Slave-Ereigniszähler über Daten-Diagnosecode 0x000A zurücksetzen  1 Wort pro Anforderung	
3 bis 10, 12 bis 79, 82 bis 255			Reserviert	

<sup>1</sup> Bei der erweiterten Adressierung ist die maximale Datenlänge je nach Datentyp der Funktion um 1 Byte oder 1 Wort geringer.

## Parameter DATA\_PTR

Der Parameter DATA\_PTR zeigt auf die DB- oder M-Adresse, in die geschrieben oder aus der gelesen wird. Wenn Sie einen Datenbaustein verwenden, müssen Sie einen globalen Datenbaustein anlegen, der den Datenspeicher für Lese- und Schreibvorgänge auf Modbus-Slaves bereitstellt.

---

### Hinweis

#### Der Datenbaustein DATA\_PTR muss die direkte Adressierung zulassen

Der Datenbaustein muss die direkte (absolute) und die symbolische Adressierung zulassen. Wenn Sie den Datenbaustein anlegen, müssen Sie das Zugriffsattribut "Standard" auswählen.

---

## Datenbausteinstrukturen für den Parameter DATA\_PTR

- Diese Datentypen gelten für das **Lesen von Wörtern** der Modbus-Adressen 30001 bis 39999, 40001 bis 49999 und 400001 bis 465536 sowie für das **Schreiben von Wörtern** in die Modbus-Adressen 40001 bis 49999 und 400001 bis 465536.
  - Standard-Array der Datentypen WORD, UINT oder INT
  - Benannte Struktur vom Typ WORD, UINT oder INT, bei der jedes Element einen eindeutigen Namen und einen 16-Bit-Datentyp hat.
  - Benannte komplexe Struktur, bei der jedes Element einen eindeutigen Namen und einen 16- oder 32-Bit-Datentyp hat.
- Zum **Lesen** und Schreiben von Bits für die Modbus-Adressen 00001 bis 09999 und zum Lesen von Bits von 10001 bis 19999.
  - Standardfeld aus Booleschen Datentypen.
  - Benannte Boolesche Struktur aus eindeutig benannten Booleschen Variablen.
- Es ist zwar nicht erforderlich, doch empfehlenswert, dass jede Anweisung MB\_MASTER einen eigenen getrennten Speicherbereich hat. Grund hierfür ist, dass die Möglichkeit der Datenzerstörung größer ist, wenn mehrere Anweisungen MB\_MASTER in demselben Speicherbereich lesen und schreiben.
- Es ist nicht erforderlich, dass die Datenbereiche für DATA\_PTR in demselben globalen Datenbaustein liegen. Sie können einen Datenbaustein mit mehreren Bereichen für Modbus-Lesevorgänge, einen Datenbaustein für Modbus-Schreibvorgänge oder einen Datenbaustein für jede Slave-Station anlegen.

## Variablen im Datenbaustein des Modbus-Masters

Die folgende Tabelle zeigt die öffentlichen statischen Variablen im Instanz-DB von MB\_MASTER, die in Ihrem Programm verwendet werden können.

Tabelle 12- 70 Statische Variablen im Instanz-DB

Variable	Datentyp	Anfangswert	Beschreibung
Blocked_Proc_Timeout	Real	3.0	Zeitdauer (in Sekunden), die auf eine blockierte Modbus-Master-Instanz gewartet werden soll, bevor diese Instanz als AKTIV entfernt wird. Dies kann beispielsweise vorkommen, wenn eine Master-Anforderung ausgegeben wurde und das Programm dann aufhört, die Masterfunktion aufzurufen, bevor es die Anforderung vollständig beendet hat. Der Zeitwert muss größer als 0 und kleiner als 55 Sekunden sein, ansonsten tritt ein Fehler auf. Der Standardwert ist 0,5 Sekunden.
Extended_Addressing	Bool	Falsch	Konfiguriert die Slave-Adressierung als einzelnes oder doppeltes Byte. Der Standardwert ist 0. (0 = Ein-Byte-Adresse, 1 = Zwei-Byte-Adresse)

Ihr Programm kann in die Variablen Blocked\_Proc\_Timeout und Extended\_Addressing Werte schreiben, um die Modbus-Master-Operationen zu steuern. Im Abschnitt zur Anweisung MB\_SLAVE finden Sie in der Beschreibung der Variablen HR\_Start\_Offset und Extended\_Addressing ein Beispiel dafür, wie Sie diese Variablen im Programmiereditor nutzen, sowie weitere Einzelheiten zu erweiterter Modbus-Adressierung (Seite 689).

## BedingungsCodes

Tabelle 12- 71 BedingungsCodes für die Ausführung von MB\_MASTER (Kommunikations- und Konfigurationsfehler) <sup>1</sup>

STATUS (W#16#)	Beschreibung
0000	Kein Fehler
80C8	Slave-Timeout. Prüfen Sie Baudrate, Parität und Verdrahtung des Slave.
80D1	Der Empfänger hat eine Flusskontrolle zur Unterbrechung einer laufenden Sendung angefordert, aber die Sendung innerhalb der Wartezeit nicht wieder aktiviert. Dieser Fehler wird auch bei der Hardware-Flusskontrolle erzeugt, wenn der Empfänger innerhalb der Wartezeit kein CTS erkennt.
80D2	Die Sende-anforderung wurde abgebrochen, weil vom DCE kein DSR-Signal empfangen wird.
80E0	Die Meldung wurde beendet, weil der Empfangspuffer voll ist.
80E1	Die Meldung wurde wegen eines Paritätsfehlers beendet.
80E2	Die Meldung wurde wegen eines Telegrammfehlers beendet.
80E3	Die Meldung wurde wegen eines Überlauffehlers beendet.
80E4	Die Meldung wurde abgebrochen, weil die angegebene Länge die Gesamtpuffergröße überschreitet.
8180	Ungültiger Wert für die Port-ID oder Fehler bei der Anweisung MB_COMM_LOAD
8186	Ungültige Modbus-Stationsadresse
8188	Ungültiger Modus für die Broadcast-Anforderung angegeben
8189	Ungültiger Wert für die Datenadresse.
818A	Ungültiger Wert für die Datenlänge.

STATUS (W#16#)	Beschreibung
818B	Ungültiger Pointer auf die lokale Datenquelle bzw. das lokale Datenziel: Größe nicht richtig
818C	Ungültiger Pointer für DATA_PTR oder Blocked_Proc_Timeout ungültig: Bei dem Datenbereich muss es sich um einen DB (der die symbolische und direkte Adressierung gestattet) oder um den Speicherbereich der Merker handeln.
8200	Port ist durch Bearbeitung einer Sendeanforderung belegt.

Tabelle 12- 72 Bedingungs-codes für die Ausführung von MB\_MASTER (Modbus-Protokollfehler) <sup>1</sup>

STATUS (W#16#)	Antwortcode vom Slave	Modbus-Protokollfehler
8380	-	CRC-Fehler
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Adresse außerhalb des gültigen Bereichs von DATA_PTR
8384	Größer als 03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)
8386	-	Funktionscode in der Antwort entspricht nicht dem Code in der Anforderung.
8387	-	Der falsche Slave hat reagiert.
8388	-	Die Slave-Antwort auf eine Schreibanforderung ist inkorrekt. Die vom Slave ausgegebene Schreibanforderung entspricht nicht dem, was der Master gesendet hat.

<sup>1</sup> Neben den oben aufgeführten Fehlern der Anweisung MB\_MASTER können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

**Siehe auch**

Punkt-zu-Punkt-Operationen (Seite 600)



### 12.5.3.3 MB\_SLAVE

Tabelle 12- 73 Anweisung MB\_SLAVE

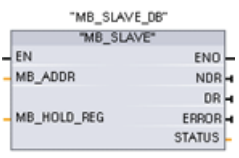
KOP/FUP	SCL	Beschreibung
	<pre>"MB_SLAVE_DB" (   MB_ADDR:=_uint_in_,   NDR=&gt;_bool_out_,   DR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   MB_HOLD_REG:=_variant_inout_);</pre>	<p>Mit der Anweisung MB_SLAVE kann Ihr Programm über einen PtP-Port eines CM (RS485 oder RS232) oder eines CB (RS485) als Modbus-Slave kommunizieren. Wenn ein dezentraler Modbus-RTU-Master eine Anforderung ausgibt, reagiert Ihr Anwenderprogramm auf die Anforderung, indem es die Anweisung MB_SLAVE ausführt. STEP 7 erstellt automatisch einen Instanz-DB, wenn Sie die Anweisung einfügen. Sie geben mit diesem MB_SLAVE_DB-Namen den Parameter MB_DB für die Anweisung MB_COMM_LOAD an.</p>

Tabelle 12- 74 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Stationsadresse des Modbus-Slaves: Standard-Adressierungsbereich (1 bis 247) Erweiterter Adressierungsbereich (0 bis 65535)
MB_HOLD_REG	IN	Variant	Pointer auf den Modbus-Halteregister-DB: Bei dem Modbus-Halteregister kann es sich um den Speicherbereich der Merker oder um einen Datenbaustein handeln.
NDR	OUT	Bool	Neue Daten bereit: <ul style="list-style-type: none"> <li>• 0 – Keine neuen Daten</li> <li>• 1 – Gibt an, dass neue Daten vom Modbus-Master geschrieben wurden</li> </ul>
DR	OUT	Bool	Daten lesen: <ul style="list-style-type: none"> <li>• 0 – Keine Daten gelesen</li> <li>• 1 – Gibt an, dass Daten vom Modbus-Master gelesen wurden</li> </ul>
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Wenn die Ausführung mit einem Fehler beendet wird, ist der Fehlercode im Parameter STATUS nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Fehlercode der Ausführung

Die Funktionscodes der Modbus-Kommunikation (1, 2, 4, 5 und 15) können Bits und Wörter direkt im Prozessabbild der Eingänge und im Prozessabbild der Ausgänge der CPU lesen und schreiben. Bei diesen Funktionscodes muss der Parameter MB\_HOLD\_REG als Datentyp größer als ein Byte definiert werden. Die folgende Tabelle zeigt die Beispielzuordnung der Modbus-Adressen zum Prozessabbild in der CPU.

Tabelle 12- 75 Zuordnung der Modbus-Adressen zum Prozessabbild

Modbus-Funktionen						S7-1200	
Codes	Funktion	Datenbereich	Adressbereich			Datenbereich	CPU-Adresse
01	Bits lesen	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
02	Bits lesen	Eingang	10001	bis	18192	Prozessabbild der Eingänge	E0.0 bis E1023.7
04	Wörter lesen	Eingang	30001	bis	30512	Prozessabbild der Eingänge	EW0 bis EW1022
05	Bit schreiben	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
15	Bits schreiben	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7

Die Funktionscodes der Modbus-Kommunikation (3, 6, 16) nutzen ein Modbus-Halteregister, bei dem es sich um einen Adressbereich im Speicherbereich der Merker oder um einen Datenbaustein handeln kann. Der Typ des Haltesregisters wird vom Parameter MB\_HOLD\_REG der Anweisung MB\_SLAVE angegeben.

**Hinweis**

**Typ des Datenbausteins MB\_HOLD\_REG**

Ein Datenbaustein mit Modbus-Halteregister muss die direkte (absolute) und die symbolische Adressierung zulassen. Wenn Sie den Datenbaustein anlegen, müssen Sie das Zugriffsattribut "Standard" auswählen.

Die folgende Tabelle zeigt Beispiele für die Zuordnung von Modbus-Adressen zum Halteregister, das für die Modbus-Funktionscodes 03 (Wörter lesen), 06 (Wort schreiben) und 16 (Wörter schreiben) verwendet wird. Die tatsächliche obere Grenze der DB-Adressen wird vom maximalen Arbeitsspeicher und vom maximalen M-Speicher des jeweiligen CPU-Modells festgelegt.

Tabelle 12- 76 Zuordnung von Modbus-Adressen zum CPU-Speicher

Modbus-Master-Adresse	Beispiele für Parameter von MB_HOLD_REG				
	MW100	DB10.DBW0	MW120	DB10.DBW50	"Rezept".Inhaltsstoff
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Rezept".Inhaltsstoff[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Rezept".Inhaltsstoff[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Rezept".Inhaltsstoff[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Rezept".Inhaltsstoff[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Rezept".Inhaltsstoff[5]

Tabelle 12- 77 Diagnosefunktionen

Modbus-Diagnosefunktionen von MB_SLAVE der S7-1200		
Codes	Teilfunktion	Beschreibung
08	0000H	Echotest Abfragedaten ausgeben: Die Anweisung MB_SLAVE gibt einem Modbus-Master das Echo eines empfangenen Datenworts zurück.
08	000AH	Kommunikationsereigniszähler löschen: Die Anweisung MB_SLAVE löscht den Kommunikationsereigniszähler, der für Modbus-Funktion 11 verwendet wird.
11		Kommunikationsereigniszähler abrufen: Die Anweisung MB_SLAVE nutzt einen internen Kommunikationsereigniszähler, um die Anzahl erfolgreicher Modbus-Lese- und Modbus-Schreibanforderungen, die an den Modbus-Slave gesendet werden, zu erfassen. Der Zähler wird bei Funktion 8, Funktion 11 und Broadcast-Anforderungen nicht hochgezählt. Außerdem wird bei Anforderungen, die zu Kommunikationsfehlern führen (z.B. Paritäts- oder CRC-Fehler), nicht hochgezählt.

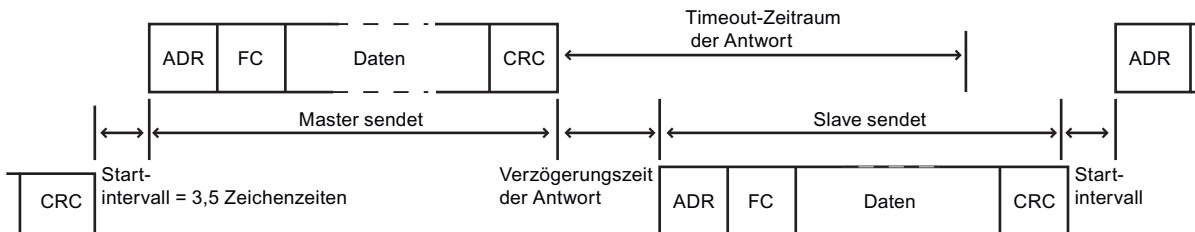
Die Anweisung MB\_SLAVE unterstützt Broadcast-Schreibanforderungen von Modbus-Mastern, solange die Anforderungen den Zugriff auf gültige Adressen umfassen. Für Funktionscodes, die bei der Broadcast-Funktion nicht unterstützt werden, erzeugt MB\_SLAVE den Fehlercode 0x8188.

### Regeln für die Modbus-Slave-Kommunikation

- MB\_COMM\_LOAD muss ausgeführt werden, um einen Port zu konfigurieren, damit die Anweisung MB\_SLAVE über diesen Port kommunizieren kann.
- Wenn ein Port als Slave auf einen Modbus-Master reagieren soll, darf dieser Port nicht mit der Anweisung MB\_MASTER programmiert werden.
- Nur eine Instanz von MB\_SLAVE kann mit einem bestimmten Port verwendet werden, andernfalls kann unerwartetes Verhalten auftreten.
- Die Modbus-Anweisungen nutzen keine Kommunikations-Alarmereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss den Kommunikationsprozess durch Abfrage der Anweisung MB\_SLAVE auf abgeschlossene Sende- und Empfangsvorgänge steuern.
- Die Anweisung MB\_SLAVE muss regelmäßig mit einer Häufigkeit ausgeführt werden, die eine zeitnahe Antwort auf eingehende Anforderungen eines Modbus-Masters ermöglicht. Es wird empfohlen, MB\_SLAVE in jedem Zyklus aus einem Programmzyklus-OB auszuführen. MB\_SLAVE kann aus einem Weckalarm-OB ausgeführt werden, doch ist dies nicht empfehlenswert, weil übermäßige Zeitverzögerungen im Interruptprogramm die Ausführung anderer Interruptprogramme temporär blockieren können.

### Zeitsteuerung des Modbus-Signals

MB\_SLAVE muss regelmäßig ausgeführt werden, um jede Anforderung des Modbus-Masters zu empfangen und entsprechend zu antworten. Die Häufigkeit der Ausführung von MB\_SLAVE richtet sich nach dem vom Modbus-Master vorgegebenen Timeout-Wert für die Antwort. Dies ist in der nachstehenden Abbildung dargestellt.



Der Timeout-Zeitraum für die Antwort RESP\_TO ist die Zeitdauer, die ein Modbus-Master auf den Beginn einer Antwort von einem Modbus-Slave wartet. Dieser Zeitraum wird nicht vom Modbus-Protokoll definiert, sondern von einem Parameter des jeweiligen Modbus-Masters. Die Häufigkeit der Ausführung (die Zeit zwischen einer Ausführung und der nächsten Ausführung) von MB\_SLAVE muss sich nach den jeweiligen Parametern des Modbus-Masters richten. Sie sollten MB\_SLAVE mindestens zweimal während des Timeout-Zeitraums für die Antwort des Modbus-Masters ausführen.

### Variablen des Modbus-Slave

Diese Tabelle zeigt die öffentlichen statischen Variablen im Instanz-Datenbaustein von MB\_SLAVE, die in Ihrem Programm verwendet werden können.

Tabelle 12- 78 Variablen des Modbus-Slave

Variable	Datentyp	Beschreibung
HR_Start_Offset	Word	Gibt die Anfangsadresse des Modbus-Halteregisters an (Standard = 0)
Extended_Adressing	Bool	Konfiguriert die Slave-Adressierung als einzelnes oder doppeltes Byte (0 = einzelne Byte-Adresse, 1 = doppelte Byte-Adresse, Standard = 0)
Request_Count	Word	Die Anzahl aller von diesem Slave empfangenen Anforderungen
Slave_Message_Count	Word	Die Anzahl der für diesen spezifischen Slave empfangenen Anforderungen
Bad_CRC_Count	Word	Die Anzahl der empfangenen Anforderungen, die einen CRC-Fehler aufweisen
Broadcast_Count	Word	Die Anzahl der empfangenen Broadcast-Anforderungen
Exception_Count	Word	Modbus-spezifische Fehler, die eine zurückgegebene Ausnahme erfordern
Success_Count	Word	Die Anzahl der für diesen spezifischen Slave empfangenen Anforderungen ohne Protokollfehler

Ihr Programm kann in die Variablen HR\_Start\_Offset und Extended\_Adressing Werte schreiben und die Modbus-Slave-Operationen steuern. Die anderen Variablen können gelesen werden, um den Modbus-Zustand zu überwachen.

## HR\_Start\_Offset

Die Adressen des Modbus-Halteregisters beginnen bei 40001 oder 400001. Diese Adressen entsprechen der Anfangsadresse des Halteregisters im Zielsystemspeicher. Sie können jedoch die Variable HR\_Start\_Offset konfigurieren, um eine andere Anfangsadresse als 40001 oder 400001 für das Modbus-Halteregister zu konfigurieren.

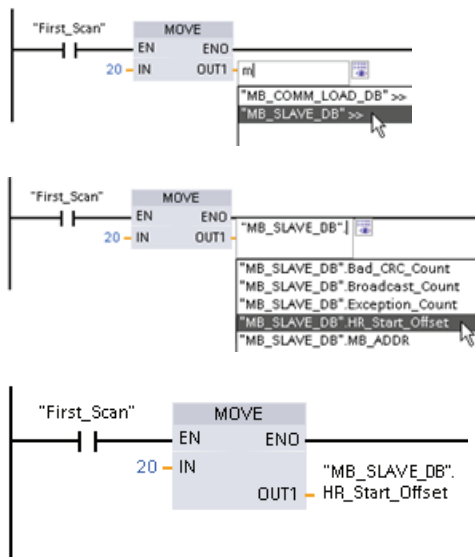
Sie können z. B. ein Halteregister mit Beginn an MW100 und einer Länge von 100 Wörtern konfigurieren. Mit einem Versatz von 20 geben Sie eine Anfangsadresse des Halteregisters von 40021 statt 40001 an. Jede Adresse unter 40021 und über 400119 führt zu einem Adressierungsfehler.

Tabelle 12- 79 Beispiel für die Adressierung des Modbus-Halteregisters

HR_Start_Offset	Adresse	Minimum	Maximum
0	Modbus-Adresse (Wort)	40001	40099
	S7-1200 Adresse	MW100	MW298
20	Modbus-Adresse (Wort)	40021	40119
	S7-1200 Adresse	MW100	MW298

HR\_Start\_Offset ist ein Wortwert, der die Anfangsadresse des Modbus-Halteregisters angibt und im Instanz-Datenbaustein MB\_SLAVE gespeichert ist. Sie können diese öffentliche statische Variable über die Parameter-Klappliste auswählen, nachdem Sie MB\_SLAVE in Ihr Programm eingefügt haben.

Wenn Sie beispielsweise MB\_SLAVE in ein KOP-Netzwerk eingefügt haben, können Sie in ein vorheriges Netzwerk gehen und den Wert HR\_Start\_Offset zuweisen. Der Wert muss vor der Ausführung von MB\_SLAVE zugewiesen werden.



Modbus-Slave-Variable über den Standard-DB-Namen eingeben:

1. Positionieren Sie den Cursor im Parameterfeld und geben Sie das Zeichen m ein.
2. Wählen Sie in der Klappliste "MB\_SLAVE\_DB" aus.
3. Positionieren Sie den Cursor rechts vom DB-Namen (nach den Anführungszeichen) und geben Sie einen Punkt ein.
4. Wählen Sie in der Klappliste "MB\_SLAVE\_DB.HR\_Start\_Offset" aus.

**Extended Addressing**

Auf die Extended Addressing-Variable wird in ähnlicher Weise zugegriffen wie auf die oben beschriebene HR\_Start\_Offset-Referenz, außer dass es sich bei der Extended Addressing-Variable um einen Booleschen Wert handelt. Der Boolesche Wert muss von einer Ausgangsspule und kann nicht von einer Box "Verschieben" geschrieben werden.

Für die Modbus-Slave-Adressierung kann ein einzelnes Byte (dies ist der Modbus-Standard) oder ein doppeltes Byte konfiguriert werden. Die erweiterte Adressierung dient zur Adressierung von mehr als 247 Geräten in einem einzigen Netzwerk. Wenn Sie sich für die erweiterte Adressierung entscheiden, können Sie maximal 64.000 Adressen ansprechen. Im Folgenden wird ein Telegramm der Modbus-Funktion 1 als Beispiel gezeigt.

Tabelle 12- 80 Slave-Adresse mit einem Byte (Byte 0)

Funktion 1	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	
Anforderung	Slave-Adresse	F-Code	Anfangsadresse		Länge der Spulen		
Gültige Antwort	Slave-Adresse	F-Code	Länge	Spulendaten			
Fehlerantwort	Slave-Adresse	0x81	E-Code				

Tabelle 12- 81 Slave-Adresse mit zwei Bytes (Byte 0 und Byte 1)

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Anforderung	Slave-Adresse		F-Code	Anfangsadresse		Länge der Spulen	
Gültige Antwort	Slave-Adresse		F-Code	Länge	Spulendaten		
Fehlerantwort	Slave-Adresse		0x81	E-Code			

**Bedingungs-codes**

Tabelle 12- 82 Bedingungs-codes für die Ausführung von MB\_SLAVE (Kommunikations- und Konfigurationsfehler) <sup>1</sup>

STATUS (W#16#)	Beschreibung
80D1	Der Empfänger hat eine Flusskontrolle zur Unterbrechung einer laufenden Sendung angefordert, aber die Sendung innerhalb der Wartezeit nicht wieder aktiviert. Dieser Fehler wird auch bei der Hardware-Flusskontrolle erzeugt, wenn der Empfänger innerhalb der Wartezeit kein CTS erkennt.
80D2	Die Sende-anforderung wurde abgebrochen, weil vom DCE kein DSR-Signal empfangen wird.
80E0	Die Meldung wurde beendet, weil der Empfangspuffer voll ist.
80E1	Die Meldung wurde wegen eines Paritätsfehlers beendet.
80E2	Die Meldung wurde wegen eines Telegrammfehlers beendet.
80E3	Die Meldung wurde wegen eines Überlauffehlers beendet.
80E4	Die Meldung wurde abgebrochen, weil die angegebene Länge die Gesamtpuffergröße überschreitet.

STATUS (W#16#)	Beschreibung
8180	Ungültiger Wert für die Port-ID oder Fehler bei der Anweisung MB_COMM_LOAD
8186	Ungültige Modbus-Stationsadresse
8187	Ungültiger Pointer auf MB_HOLD_REG-DB: Bereich ist zu klein
818C	Ungültiger Pointer MB_HOLD_REG auf M-Speicher oder DB (DB-Bereich muss symbolische und direkte Adressen zulassen)

Tabelle 12- 83 Bedingungs-codes für die Ausführung von MB\_SLAVE (Modbus-Protokollfehler) <sup>1</sup>

STATUS (W#16#)	Antwortcode vom Slave	Modbus-Protokollfehler
8380	Keine Antwort	CRC-Fehler
8381	01	Funktionscode nicht unterstützt oder nicht unterstützt in Broadcasts
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Adresse außerhalb des gültigen Bereichs von DATA_PTR
8384	03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)

<sup>1</sup> Neben den oben aufgeführten Fehlern der Anweisung MB\_SLAVE können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

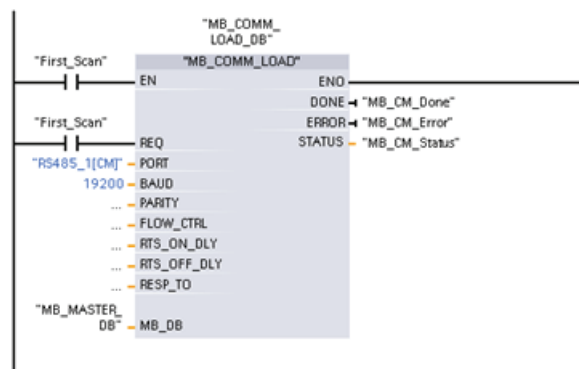
**Siehe auch**

Punkt-zu-Punkt-Operationen (Seite 600)

**12.5.3.4 Beispielprogramm für einen Modbus RTU-Master**

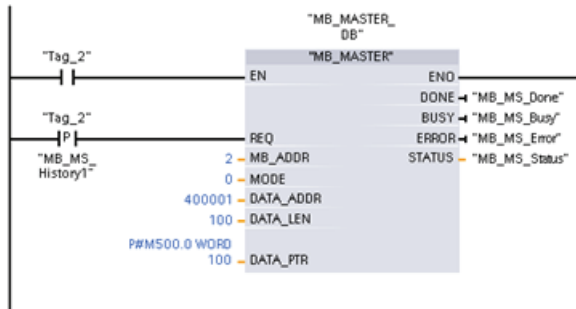
MB\_COMM\_LOAD wird während des Anlaufs über den Merker des ersten Zyklus initialisiert. Die Ausführung von MB\_COMM\_LOAD auf diese Weise darf nur durchgeführt werden, wenn sich die Konfiguration des seriellen Ports während der Laufzeit nicht ändert.

**Netzwerk 1** Parameter des RS485-Moduls nur einmal während des ersten Zyklus initialisieren.

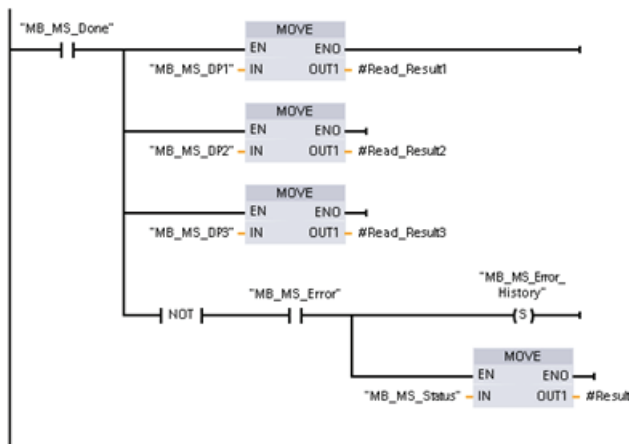


Eine Anweisung MB\_MASTER dient im Programmzyklus-OB zur Kommunikation mit einem Slave. Weitere Anweisungen MB\_MASTER können im Programmzyklus-OB für die Kommunikation mit anderen Slaves genutzt werden, oder es kann ein MB\_MASTER-FB wiederverwendet werden, um mit weiteren Slaves zu kommunizieren.

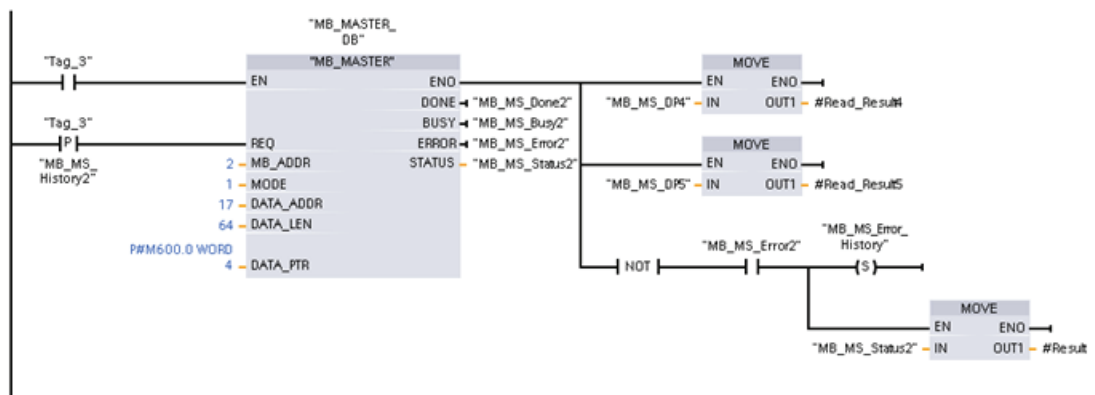
**Netzwerk 2** 100 Wörter aus dem Haltereister des Slave lesen.



**Netzwerk 3** Dies ist ein optionales Netzwerk, das die Werte der ersten 3 Wörter anzeigt, sobald die Leseoperation durchgeführt wurde.



**Netzwerk 4** 64 Bits in das Prozessabbild der Ausgänge schreiben, Beginn an Slave-Adresse A2.0.



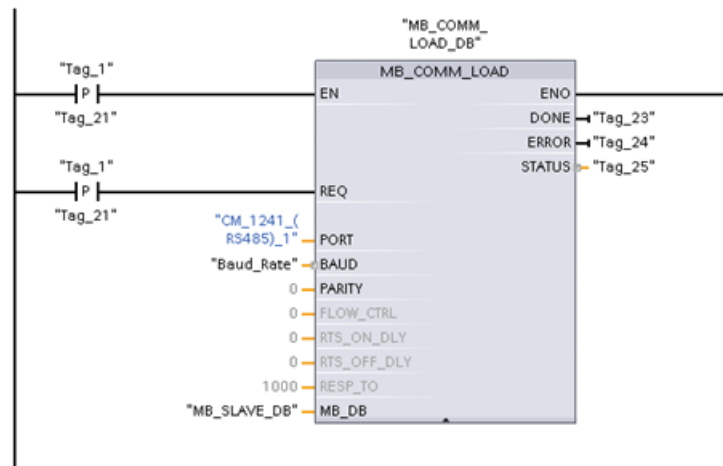


### 12.5.3.5 Beispielprogramm für einen Modbus RTU-Slave

Die im Folgenden gezeigte Anweisung MB\_COMM\_LOAD wird bei jeder Aktivierung von "Variable\_1" initialisiert.

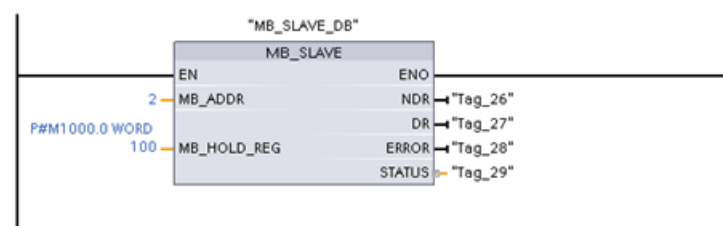
Die Ausführung von MB\_COMM\_LOAD auf diese Weise darf nur durchgeführt werden, wenn sich die Konfiguration des seriellen Ports als Folge der HMI-Konfiguration während der Laufzeit ändert.

**Netzwerk 1** RS485-Modulparameter bei jeder Änderung durch ein HMI-Gerät initialisieren



Die im Folgenden gezeigte Anweisung MB\_SLAVE befindet sich in einem zyklischen OB, der alle 10 ms ausgeführt wird. Dies bietet zwar nicht die absolut schnellste Reaktion durch den Slave, doch es bietet ein gutes Betriebsverhalten bei 9600 Baud für kurze Meldungen (bis zu 20 Byte in der Anforderung).

**Netzwerk 2** In jedem Zyklus auf Anforderungen des Modbus-Master prüfen Für das Modbus-Haltregister sind 100 Wörter mit Beginn an MW1000 konfiguriert.



## 12.6 Telecontrol und TeleService mit dem CP 1242-7

### 12.6.1 Anschluss an ein GSM-Netz

#### IP-basierte WAN-Kommunikation über GPRS

Mit Hilfe des Kommunikationsprozessors CP 1242-7 lässt sich die S7-1200 an GSM-Netze anschließen. Der CP 1242-7 ermöglicht die WAN-Kommunikation von entfernten Stationen mit einer Zentrale und die Querkommunikation zwischen Stationen.

Die Querkommunikation zwischen Stationen ist nur über das GSM-Netz möglich. Bei der Kommunikation einer entfernten Station mit einer zentralen Warte muss in der Zentrale ein PC mit Internet-Anschluss zur Verfügung stehen.

Der CP 1242-7 unterstützt folgende Dienste für die Kommunikation über das GSM-Netz:

- GPRS (General Packet Radio Service)

Der paketorientierte Dienst der Datenübertragung "GPRS" wird über das GSM-Netz abgewickelt.

- SMS (Short Message Service)

Der CP 1242-7 kann Meldungen als SMS empfangen und versenden. Kommunikationspartner kann ein Mobiltelefon oder eine S7-1200 sein.

Der CP 1242-7 ist weltweit für den industrieller Einsatz geeignet und unterstützt folgende Frequenzbänder:

- 850 MHz
- 900 MHz
- 1 800 MHz
- 1 900 MHz

## Voraussetzungen

Die Ausrüstung der Stationen oder der Zentrale hängt vom jeweiligen Anwendungsfall ab.

- Für die Kommunikation mit oder über eine zentrale Warte benötigt die Zentrale einen PC mit Internet-Anschluss.
- Für eine entfernte S7-1200-Station mit CP 1242-7, die Kommunikation über das GSM-Netz nutzen soll, sind neben der Stationsausrüstung folgende Voraussetzung erforderlich:
  - Ein Vertrag mit einem geeigneten GSM-Netzbetreiber  
Wenn GPRS genutzt werden soll, dann muss der Vertrag die Nutzung des Dienstes GPRS ermöglichen.  
Bei direkter Kommunikation zwischen Stationen nur über das GSM-Netz muss der GSM-Netzbetreiber den CPs eine feste IP-Adresse zuweisen. In diesem Fall läuft die Kommunikation zwischen den Stationen nicht über die Zentrale.
  - Die zum Vertrag gehörende SIM-Karte  
Die SIM-Karte wird in den CP 1242-7 gesteckt.
  - Lokale Verfügbarkeit eines GSM-Netzes im Bereich der Station

### 12.6.2 Anwendungen des CP 1242-7

Folgende Anwendungsfälle sind für den CP 1242-7 möglich:

## Telecontrol-Anwendungen

- Versenden von Meldungen per SMS

Über den CP 1242-7 empfängt die CPU einer entfernten S7-1200-Station SMS-Nachrichten aus dem GSM-Netz oder verschickt Meldungen per SMS an ein projektiertes Mobiltelefon oder eine S7-1200.

- Kommunikation mit einer Leitzentrale

Entfernte S7-1200-Stationen kommunizieren über das GSM-Netz und das Internet mit einem Telecontrol-Server in der Zentrale. Für die Datenübertragung per GPRS ist auf dem Telecontrol-Server in der Zentrale die Applikation "TELECONTROL SERVER BASIC" installiert. Über die integrierte OPC-Server-Funktion kommuniziert der Telecontrol-Server mit einem übergeordneten zentralen Leitsystem.

- Kommunikation zwischen S7-1200-Stationen über ein GSM-Netz

Die Kommunikation zwischen entfernten Stationen mit CP 1242-7 kann auf zwei unterschiedliche Arten abgewickelt werden:

- Querkommunikation über eine Zentrale

In dieser Konfiguration wird eine permanente gesicherte Verbindung zwischen miteinander kommunizierenden S7-1200-Stationen und dem Telecontrol-Server in der Zentrale aufgebaut. Die Kommunikation zwischen den Stationen läuft immer über den Telecontrol-Server. Der CP 1242-7 arbeitet in der Betriebsart "Telecontrol".

- Direkte Kommunikation zwischen den Stationen

Für die direkte Kommunikation zwischen Stationen ohne den Umweg über eine Zentrale werden SIM-Karten mit fester IP-Adresse eingesetzt, die es ermöglichen, die Stationen direkt zu adressieren. Die möglichen Kommunikationsdienste und Sicherheitsfunktionen (z. B. VPN) hängen dabei vom Angebot des Netzwerkbetreibers ab. Der CP 1242-7 arbeitet in der Betriebsart "GPRS direkt".

## TeleService über GPRS

Zwischen einer Engineering-Station mit STEP 7 und einer entfernten S7-1200-Station mit CP 1242-7 kann eine TeleService-Verbindung über das GSM-Netz und das Internet aufgebaut werden. Die Verbindung läuft von der Engineering-Station über einen Telecontrol-Server oder über ein TeleService-Gateway, die als Vermittler die Telegramme weiterleiten und die Autorisierung durchführen. Diese PCs nutzen die Funktionen der Applikation "TELECONTROL SERVER BASIC".

Die TeleService-Verbindung können Sie für folgende Zwecke nutzen:

- Laden von Projektierungs- oder Programmdateien aus dem STEP 7-Projekt in die Station
- Abfragen von Diagnosedaten aus der Station

### 12.6.3 Weitere Eigenschaften des CP

#### Weitere Dienste und Funktionen des CP 1242-7

- Uhrzeitsynchronisation des CP über Internet  
Die Uhrzeit des CP können Sie folgendermaßen stellen:
  - In der Betriebsart "Telecontrol" wird die Uhrzeit vom Telecontrol-Server übertragen. Der CP stellt damit seine Uhrzeit.
  - In der Betriebsart "GPRS direkt" kann der CP die Uhrzeit über SNTP anfordern.Zur Synchronisation der CPU-Uhrzeit können Sie die aktuelle Uhrzeit mithilfe eines Bausteins aus dem CP auslesen.
- Zwischenspeicherung der zu sendenden Telegramme bei Verbindungsproblemen
- Erhöhte Verfügbarkeit durch die Möglichkeit zum Verbinden mit einem Ersatz-Telecontrol-Server
- Optimiertes Datenvolumen (temporäre Verbindung)  
Alternativ zu einer permanenten Verbindung zum Telecontrol-Server kann der CP in STEP 7 mit einer temporären Verbindung zum Telecontrol-Server projektiert werden. In diesem Fall wird eine Verbindung zum Telecontrol-Server nur bei Bedarf aufgebaut.
- Protokollierung des Datenvolumens  
Die übertragenen Datenvolumina werden protokolliert und können zu weiteren Zwecken ausgewertet werden.

#### Projektierung und Baugruppentausch

Für die Projektierung der Baugruppe ist folgendes Projektierungswerkzeug erforderlich:

STEP 7 Version V11.0 SP1 oder höher

Für STEP 7 V11.0 SP1 benötigen Sie zusätzlich das Support Package "CP 1242-7" (HSP0003001).

Für die Prozessdatenübertragung per GPRS verwenden Sie im Anwenderprogramm der Station die Telecontrol-Kommunikationsanweisungen.

Die Projektierungsdaten des CP 1242-7 werden auf der jeweils lokalen CPU gespeichert. Dies ermöglicht im Ersatzteilstfall den einfachen Austausch des CP.

Sie können bis zu drei Baugruppen des Typs CP 1242-7 pro S7-1200 stecken. Damit lassen sich beispielsweise redundante Kommunikationspfade aufbauen.

### Elektrische Anschlüsse

- Spannungsversorgung des CP 1242-7  
Der CP hat einen eigenen Anschluss für die externe Spannungsversorgung DC 24 V.
- Funk-Schnittstelle für das GSM-Netz  
Für die GSM-Kommunikation ist eine externe Antenne erforderlich. Diese wird über die SMA-Buchse des CP angeschlossen.

### Weitergehende Informationen

Detaillierte Informationen enthält das Handbuch des CP 1242-7. Dieses finden Sie im Internet auf den Seiten des Siemens Industrial Automation Customer Support unter der folgenden Beitrags-ID:

42330276 (<http://support.automation.siemens.com/WW/view/de/42330276>)

## 12.6.4 Zubehör

### Die GSM/GPRS-Antenne ANT794-4MR

Für den Einsatz in GSM/GPRS-Netzen stehen folgende Antennen zur Montage im Innen- oder Außenbereich zur Verfügung:

- Quadband-Antenne ANT794-4MR



Bild 12-1 GSM/GPRS-Antenne ANT794-4MR

Kurzbezeichnung	Bestell-Nr.	Erläuterung
ANT794-4MR	6NH9 860-1AA00	Quadband-Antenne (900, 1800/1900 MHz, UMTS); witterungsbeständig für Innen- und Außenbereich; 5 m Anschlusskabel fest mit der Antenne verbunden; SMA-Stecker; inkl. Montagewinkel, Schrauben, Dübel

- Flachantenne ANT794-3M

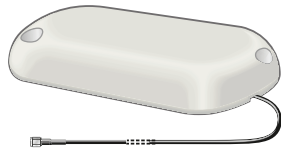


Bild 12-2 Flachantenne ANT794-3M

Kurzbezeichnung	Bestell-Nr.	Erläuterung
ANT794-3M	6NH9 870-1AA00	Flachantenne (900, 1800/1900 MHz); witterungsbeständig für Innen- und Außenbereich; 1,2 m Anschlusskabel fest mit der Antenne verbunden; SMA-Stecker; inkl. Klebepad, Schraubbefestigung möglich

Die Antennen sind separat zu bestellen.

### Weitergehende Informationen

Detaillierte Informationen finden Sie im Handbuch des Geräts. Dieses finden Sie im Internet auf den Seiten des Siemens Industrial Automation Customer Support unter folgender ID:

23119005 (<http://support.automation.siemens.com/WW/view/de/23119005>)

## 12.6.5 Konfigurationsbeispiele für Telecontrol

Im Folgenden finden Sie einige Konfigurationsbeispiele für Stationen mit CP 1242-7.

### SMS-Versand

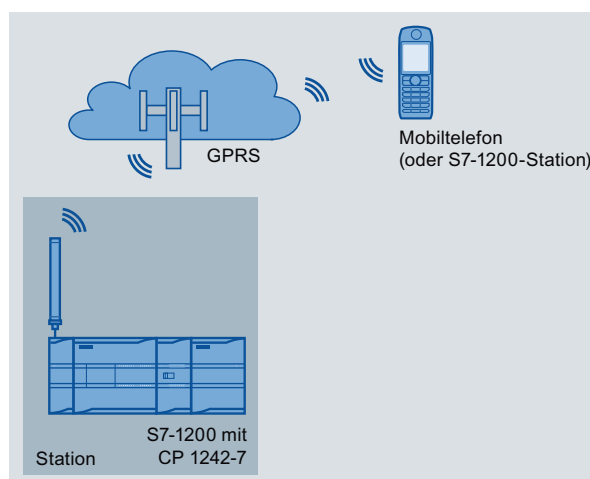


Bild 12-3 SMS-Versand einer S7-1200-Station

Eine SIMATIC S7-1200 mit CP 1242-7 kann Meldungen per SMS an ein Mobiltelefon oder eine projektierte S7-1200-Station versenden.

### Telecontrol durch eine Zentrale

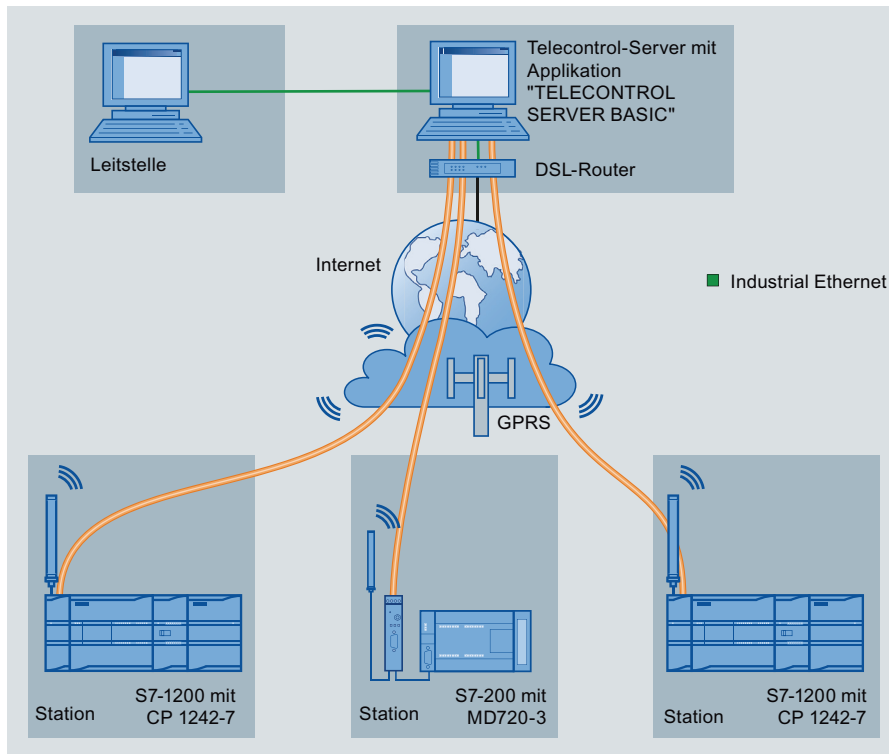


Bild 12-4 Kommunikation von S7-1200-Stationen mit einer Zentrale



Bei den Telecontrol-Anwendungen kommunizieren SIMATIC S7-1200-Stationen mit CP 1242-7 über das GSM-Netz und Internet mit einer Zentrale. Auf dem Telecontrol-Server in der Zentrale ist die Applikation "TELECONTROL SERVER BASIC" (TCSB) installiert. Damit ergeben sich folgende Anwendungsfälle:

- Telecontrol-Kommunikation zwischen Station und Zentrale

In diesem Anwendungsfall werden Daten aus dem Feld von den Stationen über das GSM-Netz und Internet an den Telecontrol-Server in der Zentrale gesendet. Der Telecontrol-Server dient der Überwachung der entfernten Stationen.

- Kommunikation zwischen Station und einer Leitzentrale mit OPC-Client

Wie im ersten Fall kommunizieren die Stationen mit dem Telecontrol-Server. Mithilfe des integrierten OPC-Servers tauscht der Telecontrol-Server die Daten mit dem OPC-Client der Leitzentrale aus.

OPC-Client und Telecontrol-Server können sich auch auf einem einzigen Rechner befinden, beispielsweise wenn TCSB auf einem Leitstellenrechner mit WinCC installiert wird.

- Querkommunikation zwischen Stationen über eine Zentrale

Querkommunikation ist möglich mit S7-Stationen, die auch mit einem CP 1242-7 ausgerüstet sind.

Für die Querkommunikation zwischen Stationen leitet der Telecontrol-Server die Telegramme der Sender-Station an die Empfänger-Station weiter.

## Direkte Kommunikation zwischen Stationen

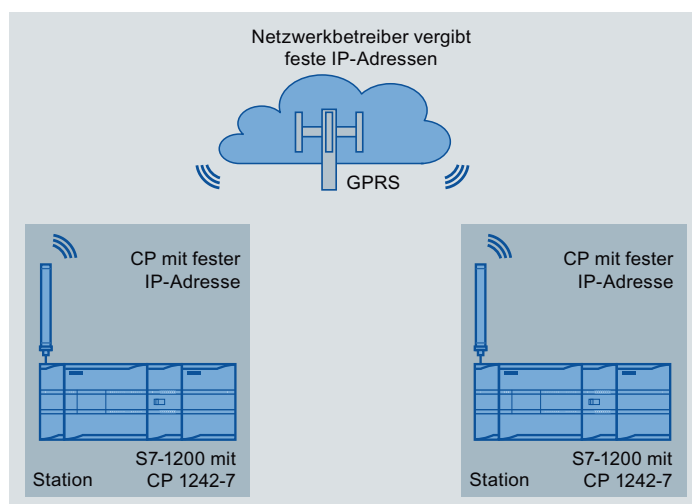


Bild 12-5 Direkte Kommunikation von zwei S7-1200-Stationen

In dieser Konfiguration kommunizieren zwei SIMATIC S7-1200-Stationen mithilfe des CP 1242-7 über das GSM-Netz direkt miteinander. Jeder CP 1242-7 hat eine feste IP-Adresse. Der entsprechende Dienst des GSM-Netzbetreibers muss dies ermöglichen.

### TeleService über GPRS

Bei TeleService über GPRS kommuniziert eine Engineering-Station, auf der STEP 7 installiert ist, über das GSM-Netz und das Internet mit dem CP 1242-7 in der S7-1200.

Da eine Firewall in der Regel für Verbindungsanforderungen von außen geschlossen ist, wird eine Vermittlerstation zwischen entfernter Station und Engineering-Station benötigt. Diese Vermittlerstation kann ein Telecontrol-Server oder, falls in der Konfiguration kein Telecontrol-Server vorhanden ist, ein TeleService-Gateway sein.

### TeleService mit Telecontrol-Server

Die Verbindung läuft über den Telecontrol-Server.

- Engineering-Station und Telecontrol-Server sind über Intranet (LAN) oder Internet verbunden.
- Telecontrol-Server und entfernte Station sind über das Internet und das GSM-Netz verbunden.

Engineering-Station und Telecontrol-Server können auch der gleiche Rechner sein, d. h., STEP 7 und TCSB sind auf dem gleichen Rechner installiert.

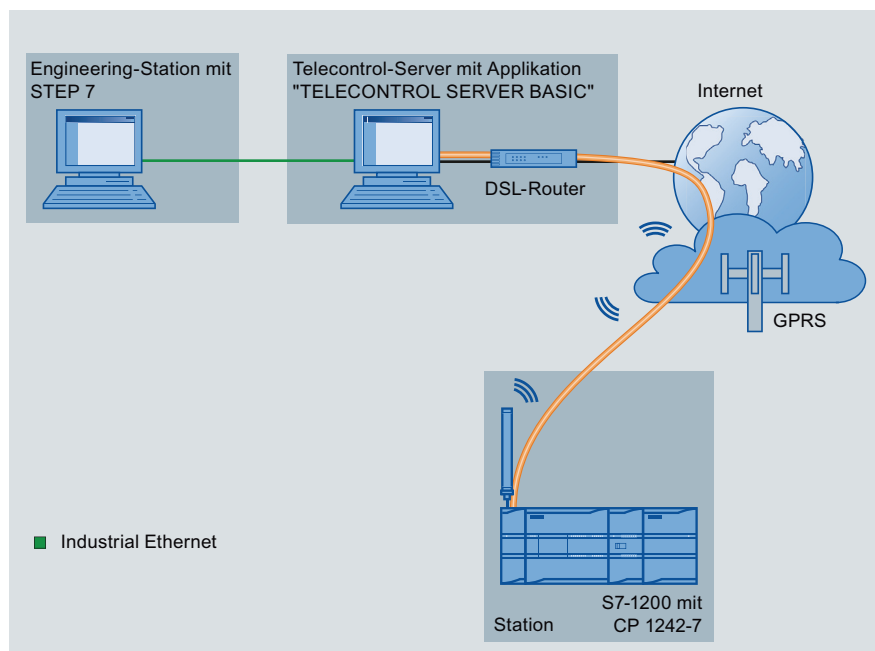


Bild 12-6 TeleService über GPRS in einer Konfiguration mit Telecontrol-Server

### TeleService ohne Telecontrol-Server

Die Verbindung läuft über das TeleService-Gateway.

Die Verbindung zwischen Engineering-Station und TeleService-Gateway kann lokal über LAN oder über das Internet laufen.

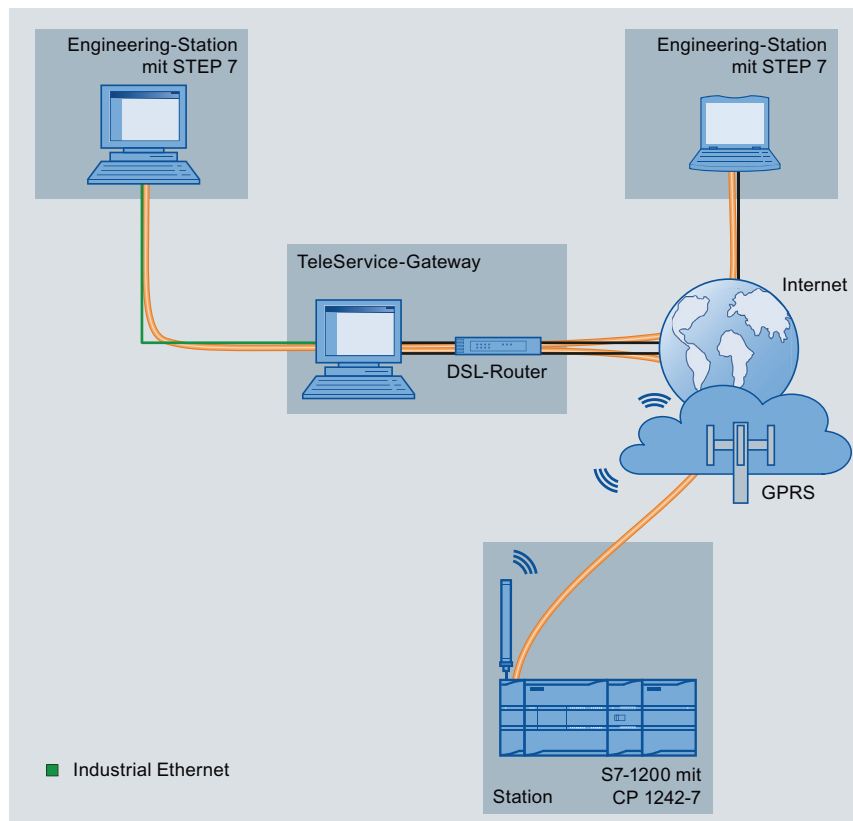


Bild 12-7 TeleService über GPRS in einer Konfiguration mit TeleService-Gateway



## Teleservice-Kommunikation (SMTP-E-Mail)

### 13.1 E-Mail-Anweisung TM\_Mail-Übertragung

Tabelle 13- 1 Anweisung TM\_MAIL

KOP/FUP	SCL	Beschreibung
<p>The diagram shows a block labeled "TM_MAIL_DB" containing a sub-block "TM_MAIL". The "TM_MAIL" block has several input and output parameters: EN, REQ, ID, TO_S, CC, SUBJECT, TEXT, and ATTACHMENT on the left; and ENO, BUSY, DONE, ERROR, and STATUS on the right. Arrows indicate the direction of data flow between these parameters.</p>	<pre>"TM_MAIL_DB" (   REQ:=_bool_in_,   ID:=_int_in_,   TO_S:=_string_in_,   CC:=_string_in_,   SUBJECT:=_string_in_,   TEXT:=_string_in_,   ATTACHMENT:=_variant_in_,   BUSY=&gt;_bool_out_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,);</pre>	<p>Die Anweisung TM_MAIL sendet mittels SMTP (Simple Mail Transfer Protocol) eine E-Mail über TCP/IP über die Industrial Ethernet-Verbindung der CPU. Wo keine Ethernet-basierte Internetverbindung verfügbar ist, kann ein optionaler Teleservice-Adapter für die Verbindung mit Telefonfestnetzleitungen verwendet werden. TM_MAIL wird asynchron ausgeführt und der Auftrag erstreckt sich über mehrere Aufrufe von TM_MAIL. Wenn Sie TM_MAIL aufrufen, müssen Sie einen Instanz-DB zuweisen. <b>Der Instanz-DB darf nicht als remanent definiert sein.</b> Dadurch wird sichergestellt, dass der Instanz-DB beim Wechsel der CPU von STOP nach RUN initialisiert wird und dass eine neue Ausführung von TM_MAIL ausgelöst werden kann.</p>

1 STEP 7 erstellt automatisch den Instanz-DB, wenn Sie die Anweisung einfügen.

Sie starten den Sendevorgang einer E-Mail mit einer positiven Flanke von 0 nach 1 am Eingangsparameter REQ. Die folgende Tabelle zeigt die Beziehungen zwischen BUSY, DONE und ERROR. Sie können den Fortschritt der Ausführung von TM\_MAIL überwachen und die Fertigstellung erkennen, indem Sie diese Parameter bei aufeinander folgenden Aufrufen auswerten.

Die Ausgangsparameter DONE, ERROR, STATUS, und SFC\_STATUS gelten nur für jeweils einen Zyklus, wenn der Zustand des Ausgangsparameters BUSY von 1 nach 0 wechselt. Ihre Programmlogik muss temporäre Ausgangszustandswerte speichern, damit Sie in nachfolgenden Programmausführungszyklen Zustandsänderungen erkennen können.

Tabelle 13- 2 Interaktion zwischen den Parametern Done, Busy und Error

DONE	BUSY	ERROR	Beschreibung
Irrelevant	1	Irrelevant	Auftrag ist in Bearbeitung.
1	0	0	Der Auftrag wurde erfolgreich ausgeführt.
0	0	1	Auftrag wurde mit Fehler beendet. Die Ursache des Fehlers finden Sie im Parameter STATUS.
0	0	0	Kein Auftrag in Bearbeitung.

Wenn die CPU während der Ausführung von TM\_MAIL in STOP versetzt wird, wird die Kommunikationsverbindung zum E-Mailserver beendet. Die Kommunikationsverbindung zum E-Mailserver geht auch verloren, wenn bei der Kommunikation der CPU auf dem Industrial Ethernet-Bus Probleme auftreten. In diesen Fällen wird der Sendevorgang unterbrochen und die E-Mail erreicht den Empfänger nicht.

**VORSICHT**

**Anwenderprogramme ändern**

Ändern Sie die Teile Ihres Anwenderprogramms, die sich direkt auf die TM\_MAIL-Aufrufe auswirken, nur, wenn:

- Sich die CPU in STOP befindet
- Keine E-Mail gesendet wird (REQ und BUSY = 0)

Dies bezieht sich insbesondere auf das Löschen und Ersetzen von Programmbausteinen, die Aufrufe von TM\_MAIL oder Aufrufe der Instanz-DBs von TM\_MAIL.

Wenn Sie verknüpfte Programmbausteine nicht speichern, können die TPC/IP-Kommunikationsfunktionen einen undefinierten Zustand einnehmen. Nach der Übertragung eines geänderten Programmbausteins müssen Sie einen CPU-Neustart (Warmstart) oder einen Kaltstart durchführen.

**Datenkonsistenz**

Der Eingangsparameter ADDR\_MAIL\_SERVER wird gelesen, wenn der Vorgang gestartet wird. Ein neuer Wert wird erst wirksam, wenn der gegenwärtige Vorgang beendet ist und eine neue Anweisung TM\_MAIL initiiert wird.

Im Gegensatz dazu werden die Parameter WATCH\_DOG\_TIME, TO\_S, CC, FROM, SUBJECT, TEXT, ATTACHMENT, USERNAME und PASSWORD während der Ausführung von TM\_MAIL gelesen und können erst geändert werden, wenn der Auftrag beendet ist (BUSY = 0)

### Wählverbindung: IE-Parameter des TS-Adapters konfigurieren

Sie müssen die IE-Parameter des Teleservice-Adapters für ausgehende Aufrufe konfigurieren, um eine Wählverbindung zum Server Ihres Internet-Dienstanbieters herzustellen. Wenn Sie das Attribut für den Aufruf auf Anforderung einrichten, dann wird die Verbindung nur hergestellt, wenn eine E-Mail gesendet wird. Bei einer analogen Modemverbindung ist für den Verbindungsvorgang mehr Zeit erforderlich (ca. eine zusätzliche Minute). Sie müssen diese zusätzliche Zeit in den Wert WATCH\_DOG\_TIME einschließen.

Tabelle 13- 3 Datentypen für die Parameter

Parameter und Datentyp		Datentypen	Beschreibung
REQ	IN	Bool	Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet.
ID	IN	Int	Verbindungskennung: Siehe ID-Parameter der Anweisungen TCON, TDISCON, TSEND und TRCV. Eine Zahl, die nicht für weitere Instanzen dieser Anweisung im Anwenderprogramm verwendet wird, muss verwendet werden.
TO_S	IN	String	Empfängeradressen: STRING-Daten mit einer maximalen Länge von 240 Zeichen.
CC	IN	String	Empfängeradressen für CC-Kopie (optional): STRING-Daten mit einer maximalen Länge von 240 Zeichen.
SUBJECT	IN	String	Betreff der E-Mail: STRING-Daten mit einer maximalen Länge von 240 Zeichen.
TEXT	IN	String	Text der E-Mail (optional): STRING-Daten mit einer maximalen Länge von 240 Zeichen. Enthält dieser Parameter eine leere Zeichenkette, wird die E-Mail ohne Textnachricht gesendet.
ATTACHMENT	IN	Variant	Pointer auf E-Mail-Anhang: Byte-, Wort- oder Doppelwortdaten mit einer maximalen Länge von 65.534 Byte. Wird kein Wert zugewiesen, wird die E-Mail ohne Anhang gesendet.
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - Auftrag noch nicht gestartet oder wird noch ausgeführt.</li> <li>• 1 - Auftrag wurde fehlerfrei ausgeführt.</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - Kein Vorgang in Bearbeitung</li> <li>• 1- Vorgang in Bearbeitung</li> </ul>
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang = 1, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Ausgang STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = 1 ist.
STATUS	OUT	Word	Rückgabewert oder Fehlerinformationen der Anweisung TM_MAIL.

Parameter und Datentyp	Datentypen	Beschreibung
ADDR_MAIL_SERVER	1 Static	DWord
		<p>IP-Adresse des Mailserver: Sie müssen jedes Fragment der IP-Adresse als Oktett aus zwei 4-Bit-Hexadezimalzeichen zuweisen. Ist das Fragment der IP-Adresse = Dezimalwert 10, was dem Hexadezimalwert A entspricht, dann müssen Sie für dieses Oktett "0A" eingeben.</p> <p>Beispiel: IP-Adresse = 192.168.0.10</p> <p>ADDR_MAIL_SERVER = DW#16#C0A8000A, wobei gilt:</p> <ul style="list-style-type: none"> <li>• 192 = 16#C0,</li> <li>• 168 = 16#A8</li> <li>• 0 = 16#00</li> <li>• 10 = 16#0A</li> </ul>
WATCH_DOG_TIME	1 Static	Time
		<p>Der maximale Zeitraum, während dessen TM_MAIL eine Serververbindung herstellen muss. Wird dieser Zeitraum überschritten, wird die Ausführung von TM_MAIL mit einem Fehler beendet.</p> <p>Die tatsächliche Zeitverzögerung, bis TM_MAIL endet und der Fehler ausgegeben wird, kann die WATCH_DOG_TIME überschreiten, weil zum Trennen der Verbindung zusätzliche Zeit erforderlich ist.</p> <p>Sie sollten zunächst einen Zeitraum von 2 Minuten einrichten. Dieser Zeitraum kann bei einer ISDN-Telefonverbindung sehr viel kleiner sein.</p>
USERNAME	1 Static	String
		Benutzername des Mailkontos: STRING-Daten mit einer maximalen Länge von 180 Zeichen.
PASSWORD	1 Static	String
		Passwort für den Mailserver: STRING-Daten mit einer maximalen Länge von 180 Zeichen.
FROM	1 Static	String
		Adresse des Absenders: STRING mit einer maximalen Länge von 240 Zeichen
SFC_STATUS	1 Static	Word
		Bedingungscode der Ausführung der aufgerufenen Kommunikationsbausteine

<sup>1</sup> Die Werte dieser Parameter werden nicht bei jedem Aufruf von TM\_MAIL modifiziert. Die Werte werden im Instanz-Datenbaustein TM\_MAIL zugewiesen und nur einmal referenziert, im ersten Aufruf von TM\_MAIL.

### SMTP-Authentifizierung

TM\_MAIL unterstützt das SMTP-AUTH-LOGIN-Authentifizierungsverfahren. Informationen zu diesem Authentifizierungsverfahren finden Sie im Handbuch des Mailserver oder auf der Website Ihres Internet-Diensteanbieters.

Das AUTH LOGIN-Authentifizierungsverfahren verwendet die Parameter USERNAME und PASSWORD der Anweisung TM\_MAIL, um eine Verbindung zum Mailserver herzustellen. Der Benutzername und das Passwort müssen zuvor in einem E-Mail-Konto auf einem E-Mailserver eingerichtet sein.

Wenn dem Parameter USERNAME kein Wert zugewiesen ist, wird das AUTH LOGIN-Authentifizierungsverfahren nicht verwendet und die E-Mail wird ohne Authentifizierung gesendet.



### **TO\_S:, CC: und FROM: Parameter**

Die Parameter TO\_S:, CC: und FROM: sind Zeichenketten, wie in den folgenden Beispielen gezeigt:

TO\_S: <wenna@mydomain.com>, <ruby@mydomain.com> ,

CC: <admin@mydomain.com>, <judy@mydomain.com> ,

FROM: <admin@mydomain.com>

Die folgenden Regeln müssen bei der Eingabe dieser Zeichenfolgen beachtet werden:

- Die Zeichen "TO\_S:", "CC:" und "FROM:" müssen eingegeben werden, einschließlich dem Doppelpunkt.
- Ein Leerzeichen und eine öffnende spitze Klammer "<" müssen vor jeder Adresse stehen. Beispielsweise muss sich zwischen "TO\_S:" und <E-Mail-Adresse> ein Leerzeichen befinden.
- Eine abschließende spitze Klammer ">" muss nach jeder Adresse eingegeben werden.
- Bei den Adressen TO\_S: und CC: muss nach jeder Adresse ein Komma "," eingegeben werden. Beispielsweise ist das Komma nach der einzelnen E-Mail-Adresse in "TO\_S: <email address>," erforderlich.
- Für die Eingabe FROM: darf nur eine E-Mail-Adresse angegeben werden, ohne Komma am Ende.

Wegen des Laufzeitmodus und der Speicherauslastung wird für die Daten TO\_S:, CC: und FROM: der Anweisung TM\_MAIL keine Syntaxprüfung durchgeführt. Wenn die Formatregeln oben nicht exakt eingehalten werden, schlägt die Übertragung des SMTP-E-Mail-servers fehl.

### **Parameter STATUS und SFC\_STATUS**

Die von TM\_MAIL zurückgegebenen Bedingungs-codes der Ausführung können wie folgt klassifiziert werden:

- W#16#0000: Ausführung von TM\_MAIL wurde erfolgreich beendet.
- W#16#7xxx: Zustand der Ausführung von TM\_MAIL
- W#16#8xxx: Fehler in einem internen Aufruf eines Kommunikationsgeräts oder des Mail-servers.

Die folgende Tabelle zeigt die Bedingungs-codes der Ausführung von TM\_MAIL, mit Ausnahme der Bedingungs-codes von intern aufgerufenen Kommunikationsmodulen.

**Hinweis**

**Anforderungen an den E-Mailserver**

TM\_MAIL kann nur mittels SMTP über Port 25 mit einem E-Mailserver kommunizieren. Die zugewiesene Portnummer kann nicht geändert werden.

Die meisten IT-Abteilungen und externen E-Mailserver blockieren mittlerweile Port 25, um zu verhindern, dass ein mit einem Virus infizierter PC zu einem gefährlichen E-Mail-Generator wird.

Sie können über SMTP eine Verbindung mit einem internen E-Mailserver herstellen und den internen Server die aktuellen Sicherheitserweiterungen verwalten lassen, die erforderlich sind, um E-Mails über das Internet an einen externen E-Mailserver weiterzuleiten.

**Beispiel für die Konfiguration eines internen E-Mailserver**

Wenn Sie Microsoft Exchange als internen E-Mailserver nutzen, können Sie den Server so konfigurieren, dass der SMTP-Zugriff von der IP-Adresse, die dem S7-1200 Zielsystem zugewiesen ist, zulässig ist. Exchange-Verwaltungskonsolle konfigurieren: Serverkonfiguration > Hub-Transport > Empfangsconnectors > IP-Relay. Im Register "Netzwerke" gibt es das Feld "E-Mail von Remoteservern mit folgenden IP-Adressen empfangen". Hier geben Sie die IP-Adresse des PLC-Geräts ein, das die Anweisung TM\_MAIL ausführt. Für diesen Verbindungstyp ist bei einem internen Microsoft Exchange-Server keine Authentifizierung erforderlich.

**Konfiguration des E-Mailserver**

TM\_MAIL kann nur einen E-Mailserver verwenden, der die Kommunikation über Port 25, SMTP und die AUTH-LOGIN-Authentifizierung (optional) gestattet.

Richten Sie ein kompatibles Konto auf dem E-Mailserver ein, um die Remote-SMTP-Anmeldung zu akzeptieren. Dann bearbeiten Sie den Instanz-DB für TM\_MAIL und geben die Zeichenfolgen TM\_MAIL USERNAME und PASSWORD ein, über die die Verbindung mit Ihrem E-Mail-Konto authentifiziert wird.

Tabelle 13- 4 Bedingungscode

STATUS (W#16#...):	SFC_STATUS (W#16#...):	Beschreibung
0000	-	Die Ausführung von TM_MAIL wurde fehlerfrei beendet. Dieser STATUS-Code 0 gewährleistet nicht, dass tatsächlich eine E-Mail gesendet wurde (siehe Punkt 1 im auf diese Tabelle folgenden Hinweis).
7001	-	TM_MAIL ist aktiv (BUSY = 1).
7002	7002	TM_MAIL ist aktiv (BUSY = 1).
8xxx	xxxx	Die Ausführung von TM_MAIL wurde mit einem Fehler in den internen Aufrufen der Kommunikationsanweisung beendet. Weitere Informationen zum Parameter SFC_STATUS finden Sie in den Beschreibungen des Parameters STATUS der zugrunde liegenden offenen PROFINET-Benutzerkommunikationsanweisungen.

<b>STATUS (W#16#...):</b>	<b>SFC_STATUS (W#16#...):</b>	<b>Beschreibung</b>
8010	xxxx	Verbindung konnte nicht aufgebaut werden: Weitere Informationen zum Parameter SFC_STATUS finden Sie in der Beschreibung des Parameters STATUS der Anweisung TCON.
8011	xxxx	Fehler beim Senden der Daten: Weitere Informationen zum Parameter SFC_STATUS finden Sie in der Beschreibung des Parameters STATUS der Anweisung TSEND.
8012	xxxx	Fehler beim Empfangen der Daten: Weitere Informationen zum Parameter SFC_STATUS finden Sie in der Beschreibung des Parameters STATUS der Anweisung TRCV.
8013	xxxx	Verbindung konnte nicht aufgebaut werden: Weitere Informationen zum Auswerten des Parameters SFC_STATUS finden Sie in den Beschreibungen des Parameters STATUS der Anweisungen TCON und TDISCON.
8014	-	Verbindung konnte nicht aufgebaut werden: Sie haben möglicherweise eine inkorrekte IP-Adresse des Mailserver (ADDR_MAIL_SERVER) oder zu wenig Zeit (WATCH_DOG_TIME) für die Verbindung angegeben. Es ist auch möglich, dass die CPU keine Verbindung zum Netzwerk hat oder die CPU-Konfiguration inkorrekt ist.
82xx, 84xx, 85xx	-	Die Fehlermeldung kommt vom Mailserver und entspricht der Fehlernummer "8" des SMTP-Protokolls. Siehe Punkt 2 in dem auf diese Tabelle folgenden Hinweis.
8450	-	Vorgang wird nicht ausgeführt: Mailbox ist nicht verfügbar, wiederholen Sie den Vorgang später.
8451	-	Vorgang abgebrochen: Lokaler Fehler bei der Bearbeitung. Wiederholen Sie den Vorgang später.
8500	-	Fehler in der Befehlsyntax: Die Ursache kann sein, dass der E-Mailserver den LOGIN-Authentifizierungsvorgang nicht unterstützt. Prüfen Sie die Parameter von TM_MAIL. Versuchen Sie, eine E-Mail ohne Authentifizierung zu senden. Versuchen Sie, den Parameter USERNAME durch eine leere Zeichenkette zu ersetzen.
8501	-	Syntaxfehler: Inkorrekter Parameter oder inkorrektes Argument; Sie haben möglicherweise eine inkorrekte Adresse in den Parameter TO_S oder CC eingegeben.
8502	-	Befehl ist unbekannt oder nicht implementiert: Prüfen Sie Ihre Eingaben, insbesondere den Parameter FROM. Möglicherweise ist dieser unvollständig und Sie haben eines der Zeichen "@" oder "." weggelassen.
8535	-	SMTP-Authentifizierung ist unvollständig. Sie haben möglicherweise einen inkorrekten Benutzernamen oder ein inkorrektes Passwort eingegeben.
8550	-	Der Mailserver ist nicht erreichbar, oder Sie haben keine Zugriffsrechte. Sie haben möglicherweise einen inkorrekten Benutzernamen oder ein inkorrektes Passwort eingegeben oder Ihr Mailserver unterstützt keinen Zugriff über Anmeldung. Eine weitere Ursache für diesen Fehler kann eine fehlerhafte Eingabe des Domainnamens nach dem Zeichen "@" in den Parametern TO_S oder CC sein.
8552	-	Vorgang abgebrochen: Zugeordnete Speichergröße überschritten; wiederholen Sie den Vorgang später.
8554	-	Übertragung fehlgeschlagen: Wiederholen Sie den Vorgang später.

---

**Hinweis**

**Mögliche nicht gemeldete E-Mail-Übertragungsfehler**

- Die fehlerhafte Angabe einer Empfängeradresse erzeugt bei TM\_MAIL keinen STATUS-Fehler. In einem solchen Fall ist nicht gewährleistet, dass die zusätzlichen Empfänger (mit korrekten E-Mail-Adressen) die E-Mail empfangen.
  - Weitere Informationen zu SMTP-Fehlercodes finden Sie im Internet oder in der Fehlerdokumentation des Mailservers. Sie können die letzte Fehlermeldung des Mailservers auch auslesen. Die Fehlermeldung wird im Parameter buffer1 des Instanz-DBs von TM\_MAIL gespeichert.
-

## Online- und Diagnose-Tools

### 14.1 Status-LEDs

Die CPU und die E/A-Module nutzen LEDs, um Informationen über den Betriebszustand des Moduls oder der E/A zu liefern.

#### Status-LEDs an einer CPU

Die CPU bietet die folgenden Statusanzeigen:

- STOP/RUN
  - Gelbes Dauerlicht zeigt den Betriebszustand STOP an
  - Grünes Dauerlicht zeigt den Betriebszustand RUN an
  - Blinken (abwechselnd grün und gelb) zeigt an, dass die CPU in der Betriebsart STARTUP ist
- ERROR
  - Rotes Blinklicht zeigt einen Fehler an, z.B. einen internen Fehler in der CPU, einen Fehler der Memory Card oder einen Konfigurationsfehler (nicht übereinstimmende Module)
  - Rotes Dauerlicht zeigt defekte Hardware an
- Wenn Sie eine Memory Card stecken, blinkt die LED MAINT (Wartung). Die CPU wechselt dann in den Betriebszustand STOP. Nachdem die CPU in den Betriebszustand STOP gegangen ist, führen Sie eine der folgenden Funktionen durch, um die Auswertung der Memory Card zu starten:
  - Versetzen Sie die CPU in den Betriebszustand RUN
  - Führen Sie ein Urlöschen durch (MRES)
  - Schalten Sie die CPU aus und wieder ein

Den Zustand der LEDs können Sie auch mit der Anweisung LED (Seite 316) ermitteln.

Tabelle 14- 1 Status-LEDs für eine CPU

Beschreibung	STOP/RUN Gelb/Grün	ERROR Rot	MAINT Gelb
Netz aus	Aus	Aus	Aus
Anlauf, Selbsttest oder Firmware-Aktualisierung	Blinken (abwechselnd gelb und grün)	-	Aus
Betriebszustand STOP	Ein (gelb)	-	-
Betriebszustand RUN	Ein (grün)	-	-
Ziehen Sie die Memory Card	Ein (gelb)	-	Blinkt
Fehler	Ein (gelb oder grün)	Blinkt	-

14.1 Status-LEDs

Beschreibung	STOP/RUN Gelb/Grün	ERROR Rot	MAINT Gelb
Wartung erforderlich	Ein (gelb oder grün)	-	Ein
Hardware defekt	Ein (gelb)	Ein	Aus
LED-Test oder CPU-Firmware defekt	Blinken (abwechselnd gelb und grün)	Blinkt	Blinkt

Die CPU bietet auch zwei LEDs, die den Zustand der PROFINET-Kommunikation anzeigen. Öffnen Sie die untere Abdeckklappe der Klemmenleiste, um die PROFINET-LEDs zu sehen.

- Link (grün) wird eingeschaltet, um eine erfolgreiche Verbindung anzuzeigen
- Rx/Tx (gelb) wird eingeschaltet, um Übertragungsaktivität anzuzeigen

Die CPU und jedes digitale Signalmodul (SM) bieten eine I/O Channel -LED für jeden digitalen Eingang und Ausgang. I/O Channel (grün) wird ein- oder ausgeschaltet, um den Zustand des jeweiligen Eingangs oder Ausganges anzuzeigen.

Status-LEDs an einem SM

Außerdem bietet jedes digitale SM eine DIAG-LED, die den Zustand des Moduls anzeigt:

- Grün zeigt an, dass das Modul betriebsbereit ist
- Rot zeigt an, dass das Modul defekt oder nicht betriebsbereit ist

Jedes analoge SM bietet eine I/O Channel -LED für jeden der analogen Eingänge und Ausgänge.

- Grün zeigt an, dass der Kanal konfiguriert wurde und aktiv ist
- Rot zeigt einen Fehlerzustand des jeweiligen analogen Eingangs oder Ausganges an

Außerdem bietet jedes analoge SM eine DIAG-LED, die den Zustand des Moduls anzeigt:

- Grün zeigt an, dass das Modul betriebsbereit ist
- Rot zeigt an, dass das Modul defekt oder nicht betriebsbereit ist

Das SM erkennt das Vorhandensein bzw. die Abwesenheit von Modulspannung (feldseitige Spannung, sofern erforderlich).

Tabelle 14- 2 Status-LEDs für ein Signalmodul (SM)

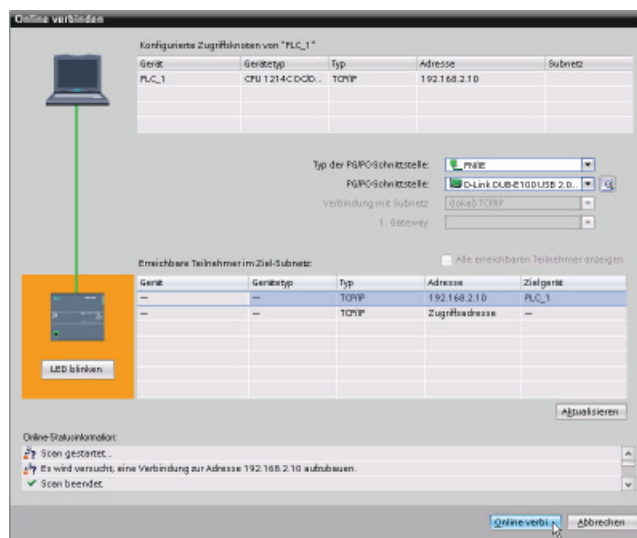
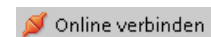
Beschreibung	DIAG (Gelb/Grün)	I/O Channel (Gelb/Grün)
Feldseitige Spannung ist aus	Rot blinkend	Rot blinkend
Nicht konfiguriert oder Aktualisierung in Bearbeitung	Grün blinkend	Aus
Modul fehlerfrei konfiguriert	Ein (grün)	Ein (grün)
Fehlerbedingung	Rot blinkend	-
E/A-Fehler (bei aktivierter Diagnose)	-	Rot blinkend
E/A-Fehler (bei deaktivierter Diagnose)	-	Ein (grün)

## 14.2 Online-Verbindung mit einer CPU herstellen

Eine Online-Verbindung zwischen dem Programmiergerät und der CPU ist zum Laden von Programmen und Projektdaten sowie für die folgenden Tätigkeiten erforderlich:

- Testen von Anwenderprogrammen
- Anzeigen und Ändern des Betriebszustands der CPU (Seite 723)
- Anzeigen und Einstellen von Datum und Uhrzeit der CPU (Seite 721)
- Anzeigen der Modulinformationen
- Vergleichen und Synchronisieren (Seite 725) von Offline- und Online-Programmbausteinen
- Laden von Programmbausteinen in die und aus der CPU
- Anzeigen von Diagnose und Diagnosepuffer (Seite 724)
- Mit einer Beobachtungstabelle (Seite 729) das Anwenderprogramm durch Beobachten und Steuern von Werten testen
- Mit einer Forcetabelle Werte in der CPU forcen (Seite 732)

Um eine Online-Verbindung zu einer konfigurierten CPU herzustellen, klicken Sie in der Projektnavigation auf die CPU und in der Projektansicht auf die Schaltfläche "Online verbinden":



Wenn Sie zum ersten Mal mit dieser CPU online gehen, müssen Sie den Typ der PG/PC-Schnittstelle und die spezifische PG/PC-Schnittstelle im Dialog "Online verbinden" auswählen, bevor Sie eine Online-Verbindung zu einer CPU auf dieser Schnittstelle herstellen.

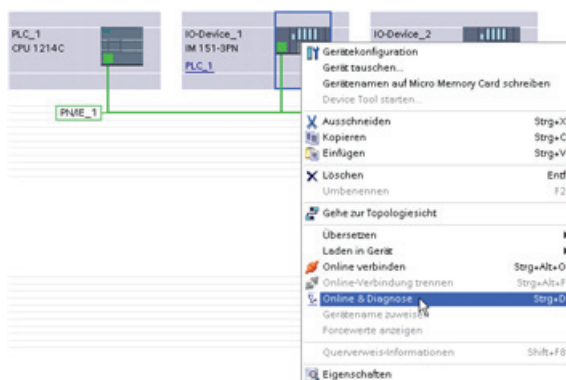
Ihr Programmiergerät ist nun mit der CPU verbunden. Orangefarbene Rahmen weisen auf eine Online-Verbindung hin. Nun können Sie die Tools unter "Online & Diagnose" in der Projektnavigation und die Taskcard "Online-Tools" verwenden.

### 14.3 Einem PROFINET IO-Gerät online einen Namen zuweisen

Den Geräten in Ihrem PROFINET-Netzwerk muss ein Name zugewiesen sein, damit Sie eine Verbindung zur CPU herstellen können. Sie weisen Ihren PROFINET-Geräten im Editor "Geräte & Netze" Namen zu, wenn die Geräte noch keinen Namen haben oder wenn der Name eines Geräts geändert werden soll.

Den Namen eines PROFINET IO-Geräts müssen Sie sowohl im STEP 7-Projekt als auch über das Werkzeug "Online & Diagnose" im Konfigurationsspeicher des PROFINET IO-Geräts zuweisen (z. B. im Konfigurationsspeicher eines ET200 S-Schnittstellenmoduls). Fehlt ein Name oder entsprechen sich die Namen an den beiden Speicherorten nicht, funktioniert der Modus für den PROFINET IO-Datenaustausch nicht.

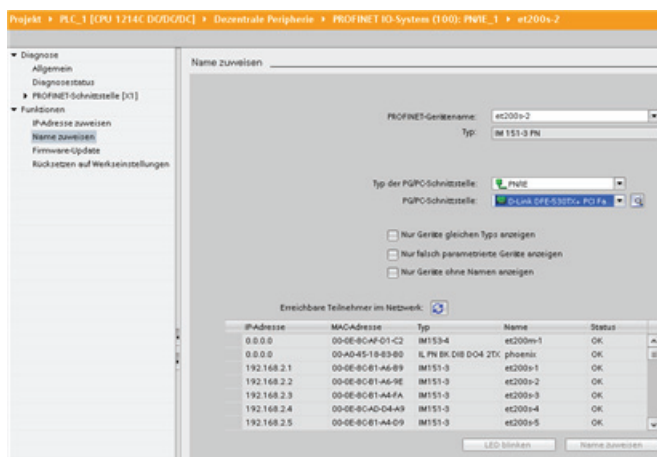
1. Klicken Sie im Editor "Geräte & Netze" mit der rechten Maustaste auf das gewünschte PROFINET IO-Gerät und wählen Sie "Online & Diagnose".



2. Wählen Sie im Dialog "Online & Diagnose" die folgenden Menübefehle:

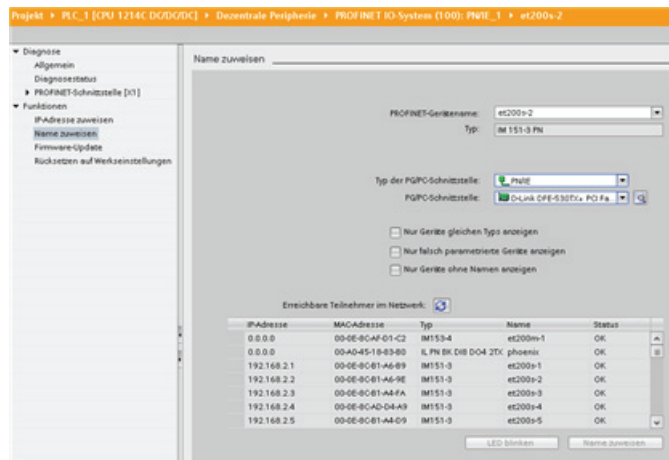
- "Funktionen"
- "Name zuweisen"

Klicken Sie auf das Symbol "Erreichbare Teilnehmer im Netzwerk", um alle PROFINET IO-Geräte im Netzwerk anzuzeigen.



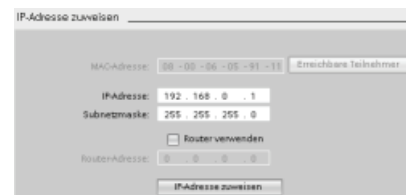
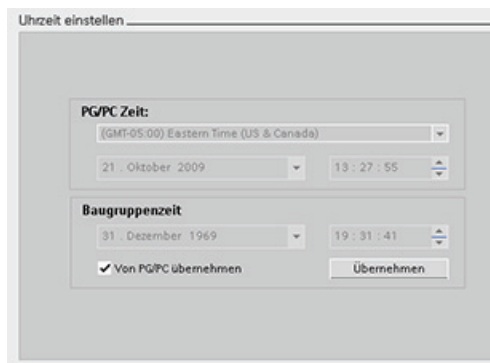


3. Klicken Sie in der daraufhin angezeigten Liste auf das gewünschte PROFINET IO-Gerät und dann auf die Schaltfläche "Name zuweisen", um den Namen in den Konfigurationsspeicher des PROFINET IO-Geräts zu schreiben.



## 14.4 Einstellen der IP-Adresse und der Uhrzeit

Sie können die IP-Adresse (Seite 145) und die Uhrzeit der Online-CPU einstellen. Nach dem Zugriff auf "Online & Diagnose" in der Projektnavigation einer Online-CPU können Sie die IP-Adresse anzeigen oder ändern. Ebenso können Sie Datum und Uhrzeit der CPU online aufrufen und ändern.



### Hinweis

Diese Funktion ist nur für eine CPU verfügbar, die entweder nur eine MAC-Adresse hat (der noch keine IP-Adresse zugewiesen wurde) oder die auf die Werkseinstellungen zurückgesetzt wurde.

## 14.5 Auf Werkseinstellungen zurücksetzen

Unter den folgenden Bedingungen können Sie eine S7-1200 auf die ursprünglichen Werkseinstellungen zurücksetzen:

- Es ist keine Memory Card in der CPU gesteckt.
- Die CPU hat eine Online-Verbindung.
- Die CPU befindet sich im Betriebszustand STOP.

---

### Hinweis

Wenn sich die CPU im Betriebszustand RUN befindet und Sie den Rücksetzvorgang starten, können Sie die CPU nach Bestätigung einer Eingabeaufforderung in den Betriebszustand STOP versetzen.

---

### Vorgehen

Um eine CPU auf die Werkseinstellungen zurückzusetzen, gehen Sie wie folgt vor:

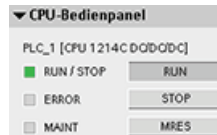
1. Öffnen Sie die Sicht "Online und Diagnose" der CPU.
2. Wählen Sie im Ordner "Funktionen" den Eintrag "Auf Werkseinstellungen zurücksetzen".
3. Aktivieren Sie das Kontrollkästchen "IP-Adresse beibehalten", wenn Sie die IP-Adresse beibehalten möchten, oder das Kontrollkästchen "IP-Adresse zurücksetzen", wenn Sie die IP-Adresse löschen möchten.
4. Klicken Sie auf die Schaltfläche "Zurücksetzen".
5. Bestätigen Sie die Eingabeaufforderung mit "OK".

### Ergebnis

Das Modul wird ggf. in den Betriebszustand STOP versetzt und auf die Werkseinstellungen zurückgesetzt:

- Der Arbeitsspeicher und interne Ladespeicher sowie alle Operandenbereiche werden gelöscht.
- Alle Parameter werden auf ihre Standardwerte zurückgesetzt.
- Der Diagnosepuffer wird gelöscht.
- Die Uhrzeit wird zurückgesetzt.
- Die IP-Adresse wird je nach vorgenommener Einstellung beibehalten oder gelöscht. (Die MAC-Adresse ist fest zugewiesen und wird niemals geändert.)

## 14.6 Bedienpanel für die Online-CPU



Im CPU-Bedienpanel wird der Betriebszustand (STOP oder RUN) der Online-CPU angezeigt. Hier wird auch angezeigt, ob ein Fehler in der CPU aufgetreten ist oder ob Werte gefordert wurden.

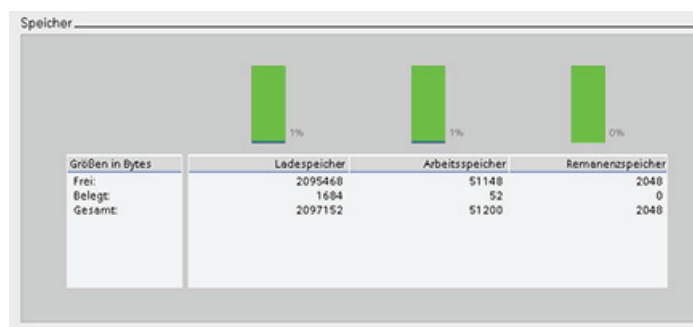
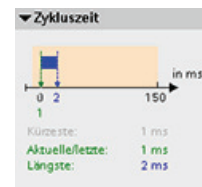
Im CPU-Bedienpanel der Taskcard "Online-Tools" können Sie den Betriebszustand einer Online-CPU ändern. Die Taskcard "Online-Tools" ist verfügbar, wenn die CPU online ist.

## 14.7 Überwachung von Zykluszeit und Speicherauslastung

Sie können die Zykluszeit und die Speicherauslastung einer Online-CPU überwachen.

Nach dem Herstellen der Verbindung zur Online-CPU öffnen Sie die Taskcard "Online-Tools", um die folgenden Messwerte anzuzeigen:

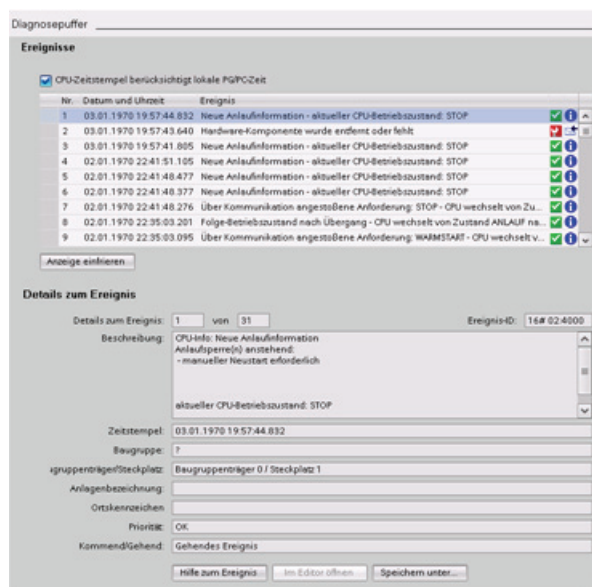
- Zykluszeit
- Speicherauslastung



## 14.8 Diagnoseereignisse in der CPU anzeigen

Im Diagnosepuffer können Sie die letzten Ereignisse in der CPU betrachten. Der Diagnosepuffer ist in der Projektnavigation über "Online & Diagnose" für eine Online-CPU verfügbar. Er enthält die folgenden Einträge:

- Diagnoseereignisse
- Änderungen im Betriebszustand der CPU (Wechsel zwischen STOP und RUN)



Der erste Eintrag entspricht dem jüngsten Ereignis. Jeder Eintrag im Diagnosepuffer enthält das Datum und die Uhrzeit, zu denen das Ereignis erfasst wurde, und eine Beschreibung.

Wie viele Einträge maximal möglich sind, hängt von der CPU ab. Es werden bis zu 50 Einträge unterstützt.

Nur die letzten 10 Ereignisse im Diagnosepuffer werden nullspannungsfest gespeichert. Wird die CPU auf die Werkseinstellungen zurückgesetzt, so werden dadurch die Einträge im Diagnosepuffer gelöscht.

Die Diagnoseinformationen können Sie auch mit der Anweisung GET\_DIAG (Seite 321) erfassen.

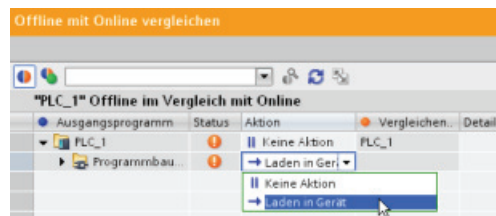
## 14.9 Vergleichen von Offline- und Online-CPU

Sie können die Codebausteine in einer Online-CPU mit den Codebausteinen Ihres Projekts vergleichen. Wenn die Codebausteine Ihres Projekts nicht den Codebausteinen der Online-CPU entsprechen, haben Sie im Editor "Vergleichen" die Möglichkeit, Ihr Projekt mit der Online-CPU abzugleichen. Laden Sie dazu entweder die Codebausteine Ihres Projekts in die CPU oder löschen Sie die Bausteine aus dem Projekt, die in der Online-CPU nicht vorhanden sind.



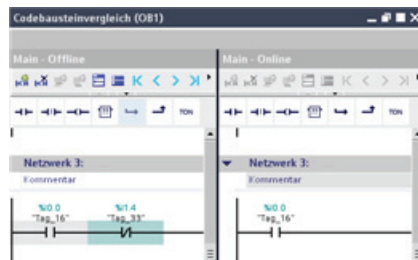
Wählen Sie die CPU in Ihrem Projekt aus.

Wählen Sie im Editor "Vergleichen" den Befehl "Offline/online vergleichen". (Rufen Sie den Befehl entweder über das Menü "Werkzeuge" oder durch Rechtsklick auf die CPU in Ihrem Projekt auf.)



Klicken Sie in die Spalte "Aktion" eines Objekts, um auszuwählen, ob das Objekt gelöscht, keine Maßnahme durchgeführt oder das Objekt in das Gerät geladen werden soll.

Durch Klicken auf die Schaltfläche "Synchronisieren" laden Sie die Codebausteine.



Klicken Sie in der Spalte "Vergleichen mit" mit der rechten Maustaste auf ein Objekt und wählen Sie "Detaillierten Vergleich starten", um die Codebausteine nebeneinander anzuzeigen.

Bei diesem Detailvergleich werden die Unterschiede zwischen den Codebausteinen der Online-CPU und den Codebausteinen der CPU in Ihrem Projekt hervorgehoben.

## 14.10 Werte in der CPU beobachten und steuern

STEP 7 bietet Online-Funktionen zum Beobachten der CPU:

- Sie können die aktuellen Werte der Variablen anzeigen oder beobachten. Die Beobachtungsfunktion ändert den Programmablauf nicht. Sie liefert Ihnen Informationen zum Programmablauf und den Daten des Programms in der CPU.
- Sie können den Ablauf und die Daten des Anwenderprogramms auch mit anderen Funktionen steuern:
  - Sie können den Wert der Variablen in der Online-CPU steuern, um zu ermitteln, wie das Anwenderprogramm reagiert.
  - Sie können einen Peripherieausgang (wie A0.1:P oder "Start":P) auf einen bestimmten Wert forcen.
  - Sie können Ausgänge im Betriebszustand STOP freischalten.

---

### Hinweis

Verwenden Sie die Steuerfunktionen stets mit Vorsicht. Diese Funktionen können die Ausführung des Anwender-/Systemprogramms schwerwiegend beeinflussen.

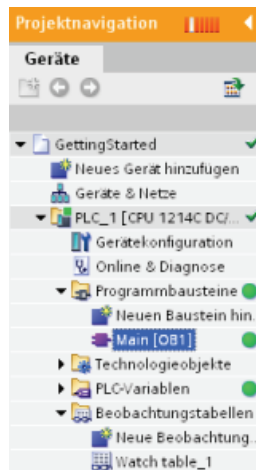
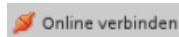
---

Tabelle 14- 3 Online-Funktionen der STEP 7-Editoren

Editor	Beobachten	Steuern	Forcen
Beobachtungstabelle	Ja	Ja	Nein
Forcetabelle	Ja	Nein	Ja
Programmieditor	Ja	Ja	Nein
Variablentabelle	Ja	Nein	Nein
DB-Editor	Ja	Nein	Nein

## 14.10.1 Online gehen, um die Werte in der CPU zu beobachten

Die Beobachtung der Variablen setzt eine Online-Verbindung zur CPU voraus. Klicken Sie in der Funktionsleiste einfach auf die Schaltfläche "Online verbinden".



Wenn Sie eine Verbindung zur CPU hergestellt haben, stellt STEP 7 die Überschriften der Arbeitsbereiche orangefarben dar.

In der Projektnavigation wird ein Vergleich des Offline-Projekts mit der Online-CPU angezeigt. Ein grüner Kreis bedeutet, dass die CPU und das Projekt synchronisiert sind, d. h. beide haben dieselbe Konfiguration und dasselbe Anwenderprogramm.

Variablen tabellen zeigen die Variablen. Beobachtungstabellen können auch die Variablen anzeigen, ebenso wie direkte Adressen.

	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
1	*On*	%E0.0	Bool		
2	*Off*	%E0.1	Bool		
3	*Run*	%A0.0	Bool		

Um die Ausführung des Anwenderprogramms zu beobachten und die Werte der Variablen anzuzeigen, klicken Sie in der Funktionsleiste auf die Schaltfläche "Alle beobachten".

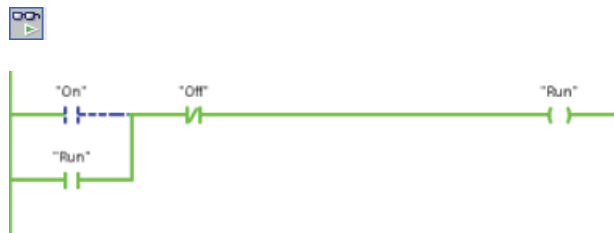
	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
1	*On*	%E0.0	Bool	<input type="checkbox"/> FALSE	
2	*Off*	%E0.1	Bool	<input type="checkbox"/> FALSE	
3	*Run*	%A0.0	Bool	<input type="checkbox"/> FALSE	

Im Feld "Beobachtungswert" wird für jede Variable der Wert angezeigt.

### 14.10.2 Zustand im Programmiereditor anzeigen

Sie können den Zustand der Variablen auch im KOP- und FUP-Editor beobachten. Rufen Sie den KOP-Editor über die Editorleiste auf. Über die Editorleiste können Sie zwischen den geöffneten Editoren umschalten, ohne die Editoren öffnen oder schließen zu müssen.

Klicken Sie im Programmiereditor in der Funktionsleiste auf die Schaltfläche "Beobachten ein/aus", um den Zustand Ihres Anwenderprogramms anzuzeigen.



Im Netzwerk im Programmiereditor wird der Signalfluss grün dargestellt.

Sie können auch mit der rechten Maustaste auf die Anweisung oder den Parameter klicken, um den Wert der Anweisung zu ändern.

### 14.10.3 Online-Werte eines DBs erfassen, um die Startwerte zurückzusetzen

Sie können die aktuellen Werte, die in einer Online-CPU beobachtet werden, erfassen und daraus Startwerte für einen globalen DB machen.

- Sie benötigen eine Online-Verbindung zur CPU.
- Die CPU muss sich im Betriebszustand RUN befinden.
- Sie müssen den DB in STEP 7 geöffnet haben.

Mit der Schaltfläche "Momentabbild der beobachteten Werte anzeigen" erfassen Sie die aktuellen Werte der ausgewählten Variablen im DB. Sie können diese Werte dann in die Spalte "Startwert" des DBs kopieren.

1. Klicken Sie im DB-Editor auf die Schaltfläche "Alle Variablen beobachten". Die Spalte "Beobachtungswert" zeigt die aktuellen Datenwerte an.
2. Klicken Sie auf die Schaltfläche "Momentabbild der beobachteten Werte anzeigen", um die aktuellen Werte in der Spalte "Momentabbild" anzuzeigen.
3. Klicken Sie auf die Schaltfläche "Alle beobachten", um die Beobachtung der Daten in der CPU zu beenden.
4. Kopieren Sie einen Wert aus der Spalte "Momentabbild" für eine Variable.
  - Wählen Sie einen zu kopierenden Wert aus.
  - Klicken Sie mit der rechten Maustaste auf den ausgewählten Wert, um das Kontextmenü aufzurufen.
  - Wählen Sie den Befehl "Kopieren".



5. Fügen Sie den kopierten Wert in die entsprechende Spalte "Startwert" der Variablen ein. (Klicken Sie mit der rechten Maustaste auf die Zelle und wählen Sie im Kontextmenü den Befehl "Einfügen".)
6. Speichern Sie das Projekt, um die kopierten Werte als neue Startwerte für den DB zu speichern.
7. Übersetzen Sie den DB und laden Sie ihn in die CPU. Der DB verwendet die neuen Startwerte, nachdem die CPU in den Betriebszustand RUN geht.

---

**Hinweis**

Die Werte, die in der Spalte "Beobachtungswert" angezeigt werden, sind immer aus der CPU kopiert. STEP 7 prüft nicht, ob alle Werte aus demselben Zyklus der CPU stammen.

---

#### 14.10.4 Werte in der CPU über die Beobachtungstabelle beobachten und steuern

Mit Hilfe einer Beobachtungstabelle können Sie die Datenpunkte beobachten und steuern, während die CPU Ihr Programm ausführt. Bei diesen Datenpunkten kann es sich je nach Beobachtungs- oder Steuerfunktion um das Prozessabbild (E oder A), um M, DB oder physische Eingänge (E\_:P) handeln. Sie können die physischen Ausgänge (A\_:P) nicht genau beobachten, weil die Beobachtungsfunktion nur den letzten geschriebenen Wert aus dem Speicherbereich A anzeigen kann und nicht den tatsächlichen Wert aus den physischen Ausgängen liest.

Die Beobachtungsfunktion ändert den Programmablauf nicht. Sie liefert Ihnen Informationen zum Programmablauf und den Daten des Programms in der CPU.

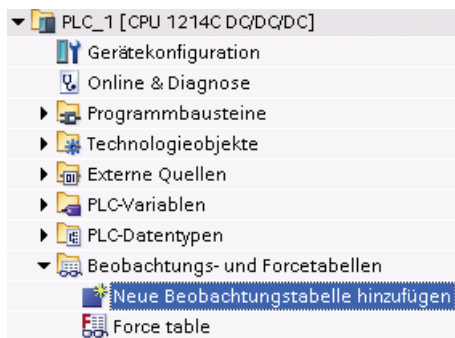
Die Steuerfunktionen ermöglichen es Ihnen, den Ablauf und die Daten des Programms zu steuern. Verwenden Sie die Steuerfunktionen mit Vorsicht. Diese Funktionen können die Ausführung des Anwender-/Systemprogramms schwerwiegend beeinflussen. Bei den drei Steuerfunktionen handelt es sich um Steuern, Forcen und Ausgänge in STOP freischalten.

Sie können mit der Beobachtungstabelle die folgenden Online-Funktionen ausführen:

- Status der Variablen beobachten
- Werte der einzelnen Variablen bearbeiten

Sie können auswählen, wann eine Variable beobachtet oder geändert werden soll:

- Zyklusbeginn: Der Wert wird bei Zyklusbeginn gelesen oder geschrieben
- Zyklusende: Der Wert wird bei Zyklusende gelesen oder geschrieben
- Umschalten nach Stop



So erstellen Sie eine Beobachtungstabelle:

1. Öffnen Sie mit Doppelklick auf "Neue Beobachtungstabelle hinzufügen" eine neue Beobachtungstabelle.
2. Geben Sie den Namen einer Variablen ein, die in der Beobachtungstabelle hinzugefügt werden soll.

Für die Beobachtung der Variablen stehen die folgenden Möglichkeiten zur Verfügung:

- Alle beobachten: Mit diesem Befehl wird die Beobachtung der sichtbaren Variablen in der aktiven Beobachtungstabelle gestartet.
- Jetzt beobachten: Mit diesem Befehl wird die Beobachtung der sichtbaren Variablen in der aktiven Beobachtungstabelle gestartet. Die Beobachtungstabelle beobachtet die Variablen sofort und nur einmal.

Zum Ändern der Variablen stehen die folgenden Möglichkeiten zur Verfügung:

- "Steuern auf 0" setzt den Wert der ausgewählten Adresse auf "0".
- "Steuern auf 1" setzt den Wert der ausgewählten Adresse auf "1".
- "Steuern jetzt" ändert den Wert der ausgewählten Adresse sofort und nur für einen Zyklus.
- "Steuern mit Trigger" ändert die Werte für die ausgewählten Adressen.

Diese Funktion erzeugt keine Rückmeldung, mit der die Änderung der ausgewählten Adressen bestätigt wird. Wird eine Bestätigung der Änderung benötigt, so ist die Funktion "Steuern jetzt" zu verwenden.

- "PA freischalten" deaktiviert den Befehl zur Sperrung von Ausgängen und steht nur zur Verfügung, wenn die CPU im Betriebszustand STOP ist.

Diese Beobachtung der Variablen setzt jedoch eine Online-Verbindung zur CPU voraus.

	Name	Adresse	Anzeigeformat	Beobachtungswert	Beobachten mit Trigger	Steuern mit Trigger	Steuervert
1	"Start"	%I0.0	Bool		Permanent	Permanent	<input type="checkbox"/>
2	"Stop"	%I0.1	Bool		Permanent	Permanent	<input type="checkbox"/>
3	"Running"	%M0.0	Bool		Permanent	Permanent	<input type="checkbox"/>

Die verschiedenen Funktionen können über die Schaltflächen am oberen Rand der Beobachtungstabelle ausgewählt werden.

Geben Sie den Namen der zu beobachtenden Variablen ein und wählen Sie in der Klappliste ein Anzeigeformat. Besteht eine Online-Verbindung zu der CPU, wird durch Anklicken der Schaltfläche "Beobachten" der Istwert der Datenpunkt im Feld "Beobachtungswert" angezeigt.

### 14.10.4.1 Variablen mit Trigger beobachten oder steuern

Der Trigger legt fest, an welchem Punkt im Zyklus die ausgewählte Adresse beobachtet oder gesteuert wird.

Tabelle 14- 4 Trigger-Arten

Trigger	Beschreibung
Permanent	Die Daten werden ständig erfasst
Zu Beginn des Zyklus	Permanent: Die Daten werden zu Beginn des Zyklus, nachdem die CPU die Eingänge gelesen hat, ständig erfasst
	Einmalig: Die Daten werden zu Beginn des Zyklus, nachdem die CPU die Eingänge gelesen hat, erfasst
Am Ende des Zyklus	Permanent: Die Daten werden am Ende des Zyklus, bevor die CPU in die Ausgänge schreibt, ständig erfasst
	Einmalig: Die Daten werden am Ende des Zyklus, bevor die CPU in die Ausgänge schreibt, erfasst
Beim Wechsel in STOP	Permanent: Die Daten werden ständig erfasst, wenn die CPU in STOP wechselt
	Einmalig: Die Daten werden einmal erfasst, nachdem die CPU in STOP wechselt

Um eine PLC-Variable mit einem bestimmten Trigger zu steuern, wählen Sie entweder den Anfang oder das Ende des Zyklus aus.

- Steuern eines Ausgangs: Das beste Triggerereignis zum Steuern eines Ausgangs ist am Ende des Zyklus, unmittelbar bevor die CPU in die Ausgänge schreibt.

Beobachten Sie den Wert der Ausgänge am Anfang des Zyklus, um zu ermitteln, welcher Wert in die physischen Ausgänge geschrieben wird. Beobachten Sie auch die Ausgänge, bevor die CPU die Werte in die physischen Ausgänge schreibt, um die Programmlogik zu prüfen und mit dem tatsächlichen E/A-Verhalten zu vergleichen.

- Steuern eines Eingangs: Das beste Triggerereignis zum Steuern eines Eingangs ist am Anfang des Zyklus, unmittelbar nachdem die CPU die Eingänge gelesen hat und bevor das Anwenderprogramm die Eingangswerte nutzt.

Wenn Sie Eingänge am Anfang des Zyklus steuern, müssen Sie auch den Wert der Eingänge am Ende des Zyklus beobachten, um sicherzustellen, dass sich der Wert des Eingangs am Ende des Zyklus nicht von dem am Anfang des Zyklus unterscheidet. Unterscheiden sich die Werte, schreibt Ihr Anwenderprogramm möglicherweise in einen falschen Eingang.

Um zu ermitteln, warum die CPU in STOP gegangen ist, erfassen Sie mit dem Trigger "Wechsel in STOP" die letzten Prozesswerte.

### 14.10.4.2 Ausgänge im Betriebszustand STOP freischalten

Mit der Beobachtungstabelle können Sie in die Ausgänge schreiben, wenn sich die CPU im Betriebszustand STOP befindet. Mithilfe dieser Funktionalität können Sie die Verdrahtung der Ausgänge prüfen und sicherstellen, dass der an eine Ausgangsklemme angeschlossene Draht das Signal 1 oder 0 an die Klemme des angeschlossenen Prozessgeräts weitergibt.



#### **WARNUNG**

Auch wenn sich die CPU im Betriebszustand STOP befindet, kann die Freischaltung eines physischen Ausgangs den an ihn angeschlossenen Prozesspunkt aktivieren.

Sie können den Zustand der Ausgänge im Betriebszustand STOP ändern, wenn die Ausgänge aktiviert sind. Sind die Ausgänge deaktiviert, können Sie die Ausgänge im Betriebszustand STOP nicht steuern.

- Um das Steuern der Ausgänge in STOP zu aktivieren, wählen Sie im Menü "Online" den Befehl "Steuern" und dann die Option "PA freischalten" oder Sie klicken mit der rechten Maustaste auf die entsprechende Zeile in der Beobachtungstabelle.

Wenn Sie dezentrale Peripherie konfiguriert haben, können Sie in der Betriebsart STOP keine Ausgänge aktivieren. Bei einem solchen Versuch wird ein Fehler gemeldet.

- Wenn die CPU in den Betriebszustand RUN wechselt, wird die Option "PA freischalten" deaktiviert.
- Wenn Eingänge oder Ausgänge geforct sind, kann die CPU im Betriebszustand STOP keine Ausgänge aktivieren. Die Force-Funktion muss zunächst beendet werden.

## 14.10.5 Werte in der CPU forcen

### 14.10.5.1 Arbeiten mit der Forcetabelle

Eine Forcetabelle bietet die Funktion "Forcen", die den Wert eines Eingangs oder Ausgangs zwangsweise auf einen vorgegebenen Wert für die Adresse des Peripherieeingangs bzw. -ausgangs setzt. Das Forcen wird im Prozessabbild der Eingänge vor der Ausführung des Anwenderprogramms und im Prozessabbild der Ausgänge vor dem Schreiben der Ausgänge in die Module durchgeführt.

---

#### **Hinweis**

Die Forcewerte werden in der CPU und nicht in der Forcetabelle gespeichert.

Sie können keinen Eingang (Adresse "E") oder Ausgang (Adresse "A") forcen. Sie können jedoch einen Peripherieeingang oder einen Peripherieausgang forcen. Die Forcetabelle hängt automatisch ein ":P" an die Adresse an (Beispiel: "On":P oder "Run":P).

---

	Name	Adresse	Anzeigeformat	Beobachtungswert	Forcewert	
1	"On":P	%I0.0:P	Bool		TRUE	<input checked="" type="checkbox"/>
2	"Off":P	%I0.1:P	Bool			<input type="checkbox"/>
3	"Run":P	%Q0.1:P	Bool			<input type="checkbox"/>

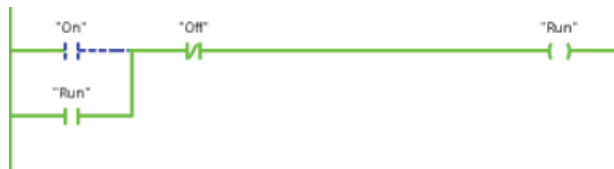
Geben Sie den Wert für den zu forcenden Eingang oder Ausgang in die Zelle "Forcewert" ein. Sie können dann das Kontrollkästchen in der Spalte "Forcen" aktivieren, um das Forcen des Eingangs oder Ausganges zu aktivieren.



Klicken Sie auf die Schaltfläche "Forcen starten oder ersetzen", um die Werte der Variablen in der Forcetable zu forcen. Klicken Sie auf die Schaltfläche "Forcen beenden", um die Werte der Variablen zurückzusetzen.

In der Forcetable können Sie den Status des geforcten Werts eines Eingangs beobachten. Den geforcten Wert eines Ausganges können Sie jedoch nicht beobachten.

Sie können den Zustand der geforcten Werte auch im Programmiereditor anzeigen.



#### ACHTUNG

Wenn ein Eingang oder Ausgang in einer Forcetable geforct wird, werden die Forceaktionen Teil der Projektkonfiguration. Beim Schließen von STEP 7 bleiben die geforcten Elemente im CPU-Programm so lange aktiv, bis sie gelöscht werden. Um diese geforcten Elemente zu löschen, müssen Sie über STEP 7 eine Verbindung zur Online-CPU herstellen und dann mithilfe der Forcetable die Forcefunktion für diese Elemente deaktivieren oder stoppen.

### 14.10.5.2 Funktionsweise der Forcefunktion

Die CPU gestattet Ihnen das Forcen von Eingängen und Ausgängen, indem Sie in der Forcetable die Adresse der physischen Eingänge und Ausgänge (E\_:P oder A\_:P) angeben und dann die Forcefunktion starten.

Im Programm werden die gelesenen Werte der physischen Eingänge durch den Forcewert überschrieben. Das Programm nutzt den geforcten Wert während der Bearbeitung. Wenn das Programm in einen physischen Ausgang schreibt, wird der Ausgangswert durch den Forcewert überschrieben. Der geforcte Wert erscheint am physischen Ausgang und wird im Prozess verwendet.

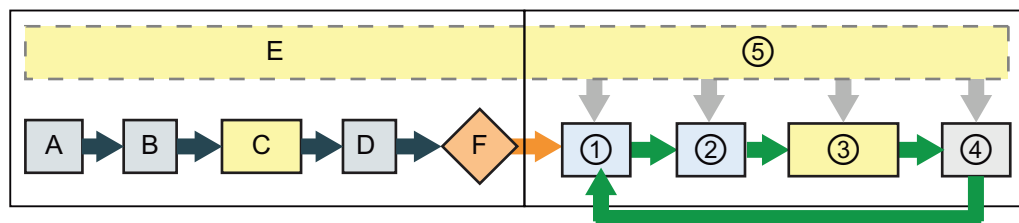
Wenn ein Eingang oder Ausgang in der Forcetable geforct wird, werden die Forceaktionen Teil des Anwenderprogramms. Auch wenn Sie die Programmiersoftware schließen, bleiben die geforcten Werte im ausgeführten CPU-Programm solange aktiv, bis Sie wieder in der Programmiersoftware online gehen und die Forcefunktion beenden. Programme mit geforcten Ein- und Ausgängen, die von einer Memory Card in eine andere CPU geladen werden, forcen auch weiterhin die im Programm ausgewählten Ein- und Ausgänge.

Wenn die CPU das Anwenderprogramm auf einer schreibgeschützten Memory Card ausführt, können Sie das Forcen von E/A nicht über eine Beobachtungstabelle auslösen oder ändern, weil Sie die Werte in dem schreibgeschützten Anwenderprogramm nicht überschreiben können. Jeder Versuch, die schreibgeschützten Werte zu forcen, führt zu einem Fehler. Bei Verwendung einer Memory Card zum Übertragen eines Anwenderprogramms werden auf dieser Memory Card gespeicherte geforcete Elemente mit an die CPU übertragen.

**Hinweis**

**Zu HSC, PWM und PTO zugewiesene digitale E/A können nicht geforcet werden**

Die vom schnellen Zähler (HSC), von der Impulsdauermodulation (PWM) und von der Impulsfolge (PTO) verwendeten E/A werden während der Konfiguration zugewiesen. Wenn diesen Funktionen digitale E/A zugewiesen werden, können die Werte der Adressen der zugewiesenen E/A nicht durch die Forcefunktion der Beobachtungstabelle geändert werden.



**Anlauf**

- A Das Löschen des Speicherbereichs E wird von der Forcefunktion nicht beeinflusst.
- B Die Initialisierung der Ausgangswerte wird von der Forcefunktion nicht beeinflusst.
- C Während der Ausführung der Anlauf-OBs schaltet die CPU den Forcewert auf, wenn das Anwenderprogramm auf den physischen Eingang zugreift.
- D Das Speichern von Alarmereignissen in der Warteschlange wird nicht beeinflusst.
- E Die Freigabe des Schreibens in die Ausgänge wird nicht beeinflusst.

**RUN**

- ① Beim Schreiben von A-Speicher in die physischen Ausgänge schaltet die CPU den Forcewert bei der Aktualisierung der Ausgänge auf.
- ② Beim Lesen der physischen Eingänge wendet die CPU die Forcewerte an, kurz bevor die Eingänge in den Speicherbereich E kopiert werden.
- ③ Während der Ausführung der Anwenderprogramms (Programmzyklus-OBs) schaltet die CPU den Forcewert auf, wenn das Anwenderprogramm auf den physischen Eingang zugreift oder in den physischen Ausgang schreibt.
- ④ Die Behandlung von Kommunikationsanforderungen und die Selbsttestdiagnose werden von der Forcefunktion nicht beeinflusst.
- ⑤ Die Verarbeitung von Alarmen während eines beliebigen Teils des Zyklus wird nicht beeinflusst.

## 14.11 Laden im Betriebszustand RUN

Die CPU unterstützt das "Laden im Betriebszustand RUN". Diese Funktion soll Ihnen ermöglichen, kleinere Änderungen am Anwenderprogramm vorzunehmen, ohne den vom Programm gesteuerten Prozess zu stören. Diese Funktion ermöglicht jedoch auch größere Programmänderungen, die den Prozess beeinträchtigen oder sogar gefährlich werden können.

### WARNUNG

Wenn Sie im Betriebszustand RUN Änderungen in die CPU laden, wirken sich die Änderungen sofort auf den Prozess aus. Wenn Sie das Programm im Betriebszustand RUN ändern, kann dies zu unerwartetem Verhalten im Prozess führen und Tod, schwere Körperverletzungen und/oder Sachschaden können die Folge sein.

Nur dazu befugtes Personal mit Kenntnis der Auswirkungen einer Programmbearbeitung in RUN auf das Prozessverhalten darf einen Ladevorgang im Betriebszustand RUN durchführen.

Die Funktion zum "Laden im Betriebszustand RUN" ermöglicht Ihnen, Änderungen an einem Programm vorzunehmen und sie in die CPU zu laden, ohne nach STOP wechseln zu müssen.

- Sie können kleinere Änderungen am aktuellen Prozess vornehmen (z. B. eine Parameterwertänderung), ohne den Prozess herunterfahren zu müssen.
- Außerdem können Sie mit dieser Funktion Programmfehler schneller beheben (z. B. die Logik für einen Schließerkontakt oder Öffnerkontakt invertieren).

Sie können die folgenden Änderungen an Programmbausteinen und Variablen vornehmen und sie im Betriebszustand RUN laden:

- Funktionen (FCs), Funktionsbausteine (FBs) und Variablentabellen erstellen, überschreiben und löschen
- Datenbausteine (DBs) erstellen und löschen; DB-Strukturänderungen können jedoch nicht überschrieben werden. Ausgangswerte des DBs können überschrieben werden. Im Betriebszustand RUN können Sie keinen Webserver-DB (Steuerung oder Fragment) laden.
- Organisationsbausteine (OBs) überschreiben; es können jedoch keine OBs erstellt oder gelöscht werden.

Maximal zehn Bausteine können gleichzeitig im Betriebszustand RUN geladen werden. Werden mehr als zehn Bausteine geladen, muss die CPU in den Betriebszustand STOP versetzt werden.

### 14.11 Laden im Betriebszustand RUN

Wenn Sie Änderungen in einen realen Prozess laden (im Unterschied zu einem simulierten Prozess wie z. B. bei der Fehlerbehebung in einem Programm), sollten Sie vor dem Laden unbedingt in Gedanken die möglichen Folgen für die Sicherheit der Maschinen und Maschinenbediener durchspielen.

---

#### Hinweis

Befindet sich die CPU im Betriebszustand RUN und wurden Programmänderungen vorgenommen, versucht STEP 7 zunächst stets, die Änderungen im Betriebszustand RUN zu laden. Möchten Sie dies verhindern, müssen Sie die CPU in STOP versetzen.

Werden die vorgenommenen Änderungen nicht von der Funktion "Laden im Betriebszustand RUN" unterstützt, werden Sie von STEP 7 aufgefordert, die CPU in den Betriebszustand STOP zu versetzen.

---

#### 14.11.1 Voraussetzungen für "Laden im Betriebszustand RUN"

Sie können Ihre Programmänderungen nur dann im Betriebszustand RUN in die CPU laden, wenn die folgenden Voraussetzungen erfüllt sind:

- Ihr Programm muss sich erfolgreich übersetzen lassen.
- Sie müssen die Kommunikation zwischen dem Programmiergerät, auf dem STEP 7 läuft, und der CPU erfolgreich aufgebaut haben.
- Ab V3.0 muss die Firmware der Ziel-CPU die Funktion "Laden im Betriebszustand RUN" unterstützen.

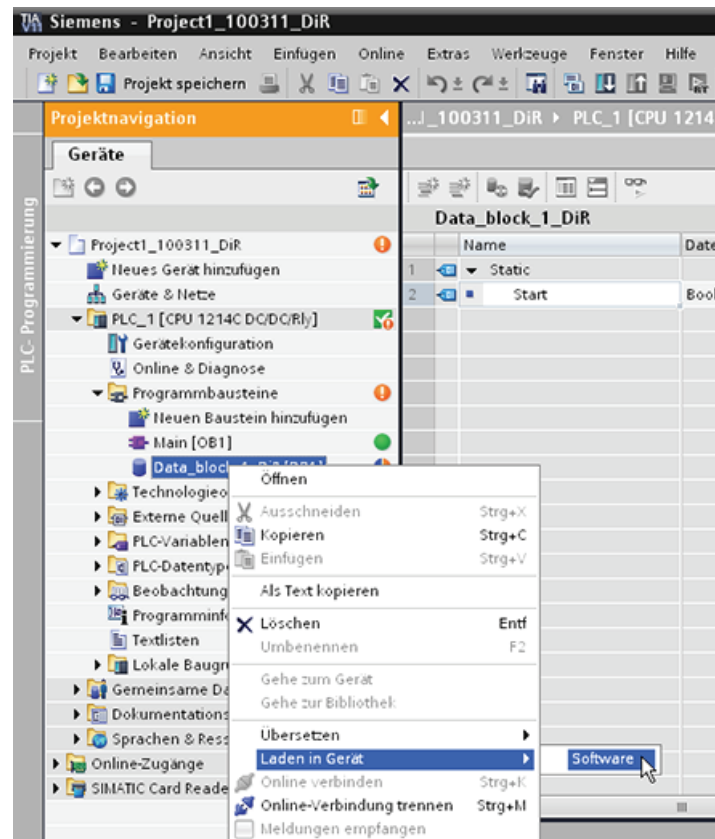
#### 14.11.2 Ändern des Programms im Betriebszustand RUN

Wenn Sie Ihr Programm im Betriebszustand RUN ändern möchten, müssen Sie zunächst sicherstellen, dass die CPU das "Laden im Betriebszustand RUN" unterstützt und sich außerdem in RUN befindet:

1. Um Ihr Programm im Betriebszustand RUN zu laden, gehen Sie auf eine der folgenden Arten vor:



- Wählen Sie im Menü "Online" den Befehl "Laden in Gerät".
- Wählen Sie in der Symbolleiste die Schaltfläche "Laden in Gerät".
- Klicken Sie in der "Projektnavigation" mit der rechten Maustaste auf "Programmbausteine" und wählen Sie den Befehl "Laden in Gerät > Software".



2. Wird das Programm erfolgreich übersetzt, lädt STEP 7 das Programm in die CPU.
3. STEP 7 fordert Sie auf, das Programm zu laden oder den Vorgang abubrechen.
4. Wenn Sie auf "Laden" klicken, lädt STEP 7 das Programm in die CPU.

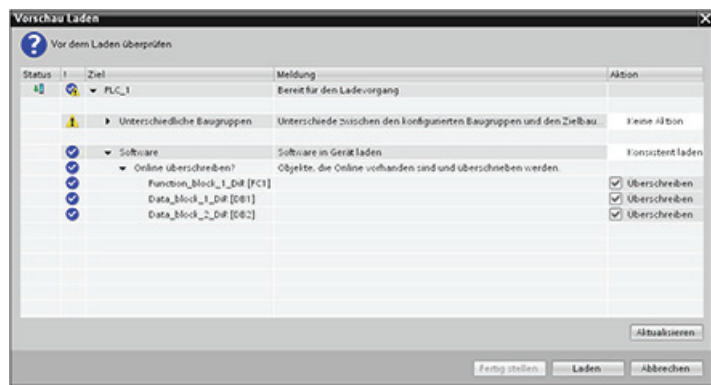
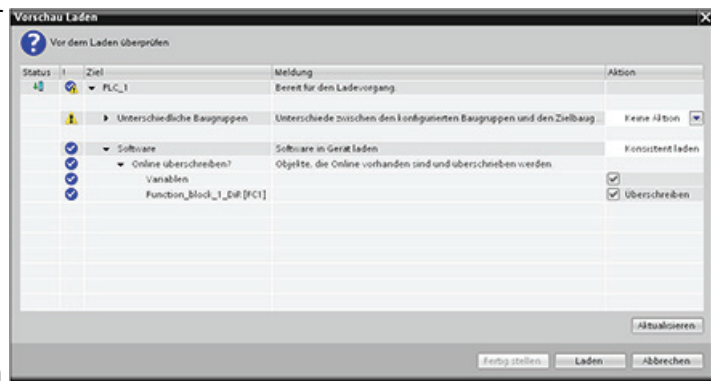
### 14.11.3 Ausgewählte Bausteine laden

Der Fokus befindet sich auf dem Ordner "Programmbausteine", auf einer Auswahl von Bausteinen oder auf einem einzelnen Baustein.

1. Wenn Sie im Bausteineditor einen einzelnen Baustein zum Laden auswählen, wird in der Spalte "Aktion" als einzige Option "Konsistent laden" angezeigt.

Sie können die Kategoriezeile erweitern, um zu prüfen, welche Bausteine geladen werden. In diesem Beispiel wurde eine geringfügige Änderung am Offline-Baustein vorgenommen und es müssen keine anderen Bausteine geladen werden.

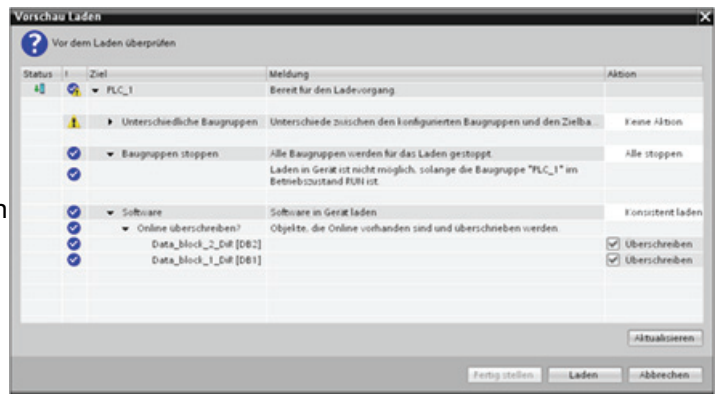
2. In diesem Beispiel müssen mehrere Bausteine geladen werden.



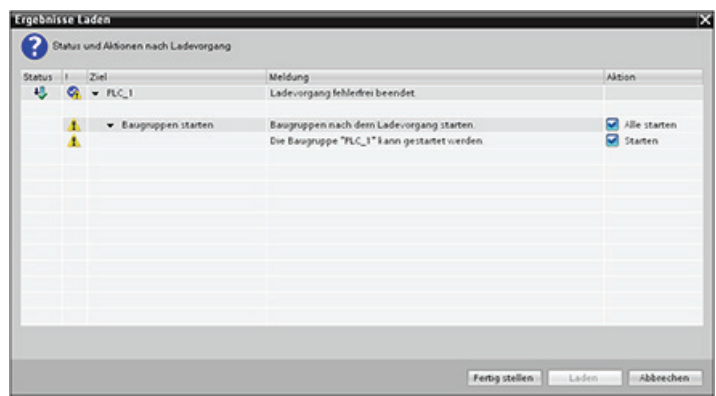
#### Hinweis

Maximal zehn Bausteine können gleichzeitig im Betriebszustand RUN geladen werden. Werden mehr als zehn Bausteine geladen, muss die CPU in den Betriebszustand STOP versetzt werden.

3. Wenn Sie versuchen, die Bausteine in RUN zu laden, doch das System erkennt vor dem tatsächlichen Ladevorgang, dass dies nicht möglich ist, erscheint im Dialog eine Zeile zum Stoppen der Module.

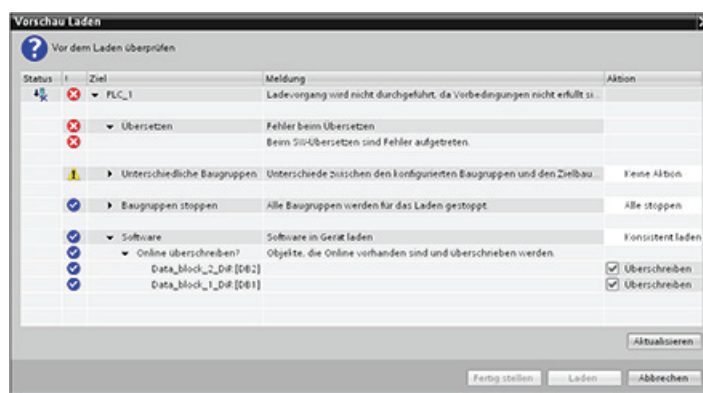


4. Klicken Sie auf die Schaltfläche "Laden". Daraufhin wird der Dialog "Ladeergebnisse" angezeigt. Klicken Sie auf die Schaltfläche "Fertig stellen", um den Ladevorgang abzuschließen.

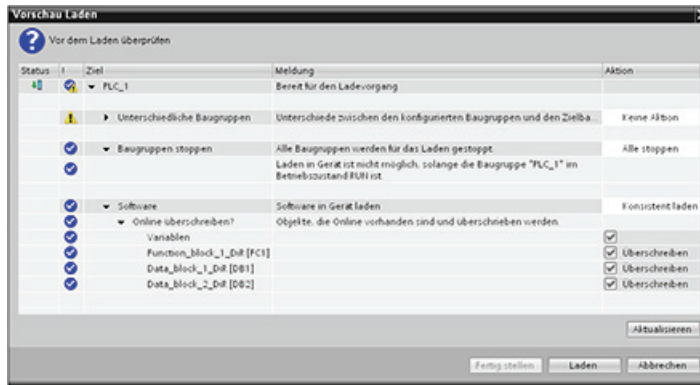


#### 14.11.4 Einen einzelnen ausgewählten Baustein mit einem Übersetzungsfehler in einem anderen Baustein laden

Wenn Sie versuchen, einen konsistenten Ladevorgang mit einem Übersetzungsfehler in einem anderen Baustein durchzuführen, zeigt der Dialog einen Fehler an und die Schaltfläche zum Laden wird deaktiviert.

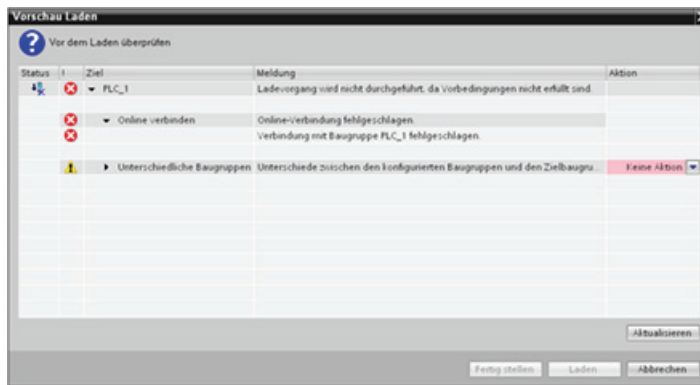


Sie müssen zunächst den Übersetzungsfehler in dem anderen Baustein korrigieren. Danach wird die Schaltfläche "Laden" wieder aktiviert.



### 14.11.5 Systemreaktion bei fehlgeschlagenem Ladevorgang

Tritt während des ersten Ladevorgangs in RUN ein Netzwerkverbindungsfehler auf, wird der im Folgenden abgebildete Dialog "Ladevorschau" angezeigt.



### 14.11.6 Laden des Programms im Betriebszustand RUN

Bevor Sie das Programm im Betriebszustand RUN laden, bedenken Sie die Auswirkungen Ihrer im Betriebszustand RUN vorgenommenen Änderungen auf den Betrieb der CPU in folgenden Fällen:

- Wenn Sie die Steuerungslogik für einen Ausgang gelöscht haben, behält die CPU den letzten Zustand des Ausganges bei, bis die CPU ausgeschaltet oder in den Betriebszustand STOP versetzt wird.
- Wenn Sie einen schnellen Zähler oder eine Impulsausgabe gelöscht haben und eine der beiden Funktionen in Betrieb ist, läuft der schnelle Zähler bzw. die Impulsausgabe bis zum nächsten Ausschalten bzw. bis zum nächsten Übergang in STOP weiter.
- Logik, die durch den Merker des ersten Zyklus aktiviert wird, wird erst nach dem nächsten Einschalten bzw. nach dem nächsten Wechsel von STOP nach RUN ausgeführt. Der Merker des ersten Zyklus wird nur durch den Wechsel in den Betriebszustand RUN gesetzt und wird nicht durch das Laden im Betriebszustand RUN beeinflusst.
- Die aktuellen Werte von Datenbausteinen (DBs) und/oder Variablen werden nicht überschrieben.

---

#### Hinweis

Damit Sie Ihr Programm im Betriebszustand RUN laden können, muss die CPU das Laden im Betriebszustand RUN unterstützen. Das Programm muss fehlerfrei übersetzt werden und die Kommunikation zwischen STEP 7 und der CPU muss fehlerfrei sein.

Sie können die folgenden Änderungen an Programmbausteinen und Variablen vornehmen und sie im Betriebszustand RUN laden:

- Funktionen (FCs), Funktionsbausteine (FBs) und Variablen tabellen erstellen, überschreiben und löschen
- Datenbausteine (DBs) erstellen und löschen; DB-Strukturänderungen können jedoch nicht überschrieben werden. Ausgangswerte des DBs können überschrieben werden. Im Betriebszustand RUN können Sie keinen Webserver-DB (Steuerung oder Fragment) laden.
- Organisationsbausteine (OBs) überschreiben; es können jedoch keine OBs erstellt oder gelöscht werden.

Maximal zehn Bausteine können gleichzeitig im Betriebszustand RUN geladen werden. Werden mehr als zehn Bausteine geladen, muss die CPU in den Betriebszustand STOP versetzt werden.

Sobald Sie einen Ladevorgang gestartet haben, können Sie erst dann weitere Vorgänge in STEP 7 durchführen, wenn der Ladevorgang beendet ist.

---

**Anweisungen, die beim "Laden im Betriebszustand RUN" fehlschlagen können**

Die folgenden Anweisungen melden möglicherweise kurzzeitige Fehler, wenn Änderungen aus dem Laden in RUN in der CPU aktiviert werden. Der Fehler tritt auf, wenn die Anweisung eingeleitet wird, während die CPU die Aktivierung der geladenen Änderungen vorbereitet. Während dieses Zeitraums unterbricht die CPU den weiteren Zugriff des Anwenderprogramms auf den Ladespeicher, solange sie noch in Bearbeitung befindliche Zugriffe des Anwenderprogramms auf den Ladespeicher abarbeitet. Dies geschieht, damit geladene Änderungen konsistent aktiviert werden können.

Anweisung	Reaktion, solange die Aktivierung ansteht
DataLogCreate	STATUS = W#16#80C0, ERROR = WAHR
DataLogOpen	STATUS = W#16#80C0, ERROR = WAHR
DataLogWrite	STATUS = W#16#80C0, ERROR = WAHR
DataLogClose	STATUS = W#16#80C0, ERROR = WAHR
DataLogNewFile	STATUS = W#16#80C0, ERROR = WAHR
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
RTM	RET_VAL = 0x80C0

In allen Fällen ist das VKE der Anweisung falsch, wenn der Fehler auftritt. Der Fehler ist temporär. Wenn er auftritt, führen Sie die Anweisung zu einem späteren Zeitpunkt aus.

**Hinweis**

Versuchen Sie nicht, die Anweisung während der aktuellen Bearbeitung des OBS auszuführen.

# Technische Daten

## A.1 Allgemeine technische Daten

### Erfüllte Normen

Der Aufbau des Automatisierungssystems S7-1200 erfüllt die folgenden Normen und Prüfvorschriften. Die Prüfkriterien für S7-1200 beruhen auf diesen Normen und Prüfvorschriften.

Beachten Sie, dass möglicherweise nicht alle S7-1200 Varianten nach diesen Normen zertifiziert sind und dass sich der Zertifizierungszustand ohne Ankündigung ändern kann. Es liegt in der Verantwortung des Anwenders, geltende Zertifizierungen anhand der auf dem Produkt angebrachten Etiketten zu ermitteln. Wenden Sie sich an Ihre Siemens-Vertretung, wenn Sie eine Liste mit den aktuellen Zulassungen für die einzelnen Bestellnummern benötigen.

### CE-Zulassung



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen und sicherheitsrelevanten Ziele der folgenden EU-Richtlinien und entspricht den harmonisierten europäischen Normen (EN) für speicherprogrammierbare Steuerungen, die in den Amtsblättern der EU aufgeführt sind.

- EU-Richtlinie 2006/95/EG (Niederspannungs-Richtlinie) "Elektrische Betriebsmittel für die Verwendung innerhalb bestimmter Spannungsgrenzen"
  - EN 61131-2:2007 Speicherprogrammierbare Steuerungen - Betriebsmittelanforderungen und Prüfungen
- EU-Richtlinie 2004/108/EG (EMV-Richtlinie) "Elektromagnetische Verträglichkeit"
  - Störaussendung  
EN 61000-6-4:2007: Industriebereich
  - Funkentstörung  
EN 61000-6-2:2005: Industriebereich
- EG-Richtlinie 94/9/EG (ATEX) "Geräte und Schutzsysteme zur bestimmungsgemäßen Verwendung in explosionsgefährdeten Bereichen"
  - EN 60079-15:2005: Schutzart 'n':

Die CE-Konformitätserklärung steht allen zuständigen Behörden zur Verfügung bei der:

Siemens AG  
IA AS RD ST PLC Amberg  
Werner-von-Siemens-Str. 50  
D92224 Amberg  
Deutschland

### cULus-Zulassung



Underwriters Laboratories, Inc. erfüllt:

- Underwriters Laboratories, Inc.: UL 508 Listed (Industriesteuerungsgeräte)
- Canadian Standards Association: CSA C22.2 Nummer 142 (Prozesssteuerungsgeräte)

#### ACHTUNG

Die Produktreihe SIMATIC S7-1200 entspricht der CSA-Norm.

Das cULus-Zeichen zeigt an, dass die S7-1200 von Underwriters Laboratories (UL) nach den Normen UL 508 und CSA 22.2 Nr. 142 geprüft und zugelassen wurde.

### FM-Zertifizierung



Factory Mutual Research (FM)

Zertifizierungsnorm Klasse Nummer 3600 und 3611

Zugelassen für den Einsatz in:

Class I, Division 2, Gas Group A, B, C, D, Temperature Class T3C Ta = 60 °C

Class I, Zone 2, IIC, Temperature Class T3 Ta = 60 °C

Canadian Class I, Zone 2 Installation nach CEC 18-150

WICHTIGE AUSNAHME: Die Anzahl von Eingängen und Ausgängen, die gleichzeitig eingeschaltet sein dürfen, finden Sie in den technischen Daten. Einige Modelle sind auf Ta = 60° C herabgesetzt.

#### ! WARNUNG

Der Austausch von Komponenten kann die Eignung für Klasse I, Division 2 und Zone 2 beeinträchtigen.

Reparatur von Geräten darf nur von einem autorisierten Siemens Service Center durchgeführt werden.

### ATEX-Zulassung



Die ATEX-Zulassung gilt nur für DC-Varianten. Die ATEX-Zulassung gilt nicht für AC- und Relaisvarianten.

EN 60079-0:2006: Explosionsfähige Atmosphäre - Allgemeine Anforderungen

EN 60079-15:2005: Elektrische Betriebsmittel für gasexplosionsgefährdete Bereiche; Schutzart 'nA'

II 3 G Ex nA II T3

WICHTIGE AUSNAHME: Die Anzahl von Eingängen und Ausgängen, die gleichzeitig eingeschaltet sein dürfen, finden Sie in den technischen Daten. Einige Modelle sind auf Ta = 60° C herabgesetzt.

Die Module in einem geeigneten Gehäuse mit einer Schutzklasse von mindestens IP54 nach EN 60529 einbauen und die Umgebungsbedingungen für den Betrieb der Geräte berücksichtigen.



Werden bei Nennbedingungen Temperaturen von 70 °C am Kabeleintritt bzw. 80 °C am Abzweigpunkt der Leitungen überschritten, so muss der zulässige Temperaturbereich des ausgewählten Kabels für die tatsächlich gemessenen Temperaturen geeignet sein.

Es sind Vorkehrungen zu treffen, um zu verhindern, dass die Nennspannung durch kurzzeitige Störungen um mehr als 40 % überschritten wird.

### C-Tick-Zulassung



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen der Normen nach AS/NZS 2064 (Klasse A).

### Koreanische Zertifizierung



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen der Koreanischen Zertifizierung (KC-Kennzeichen). Es wurde als Gerät der Klasse A eingestuft und ist für industrielle Anwendungen und nicht für die private Nutzung gedacht.

### Zulassung für das Seewesen

Die S71200 Produkte werden regelmäßig für die Zulassungen hinsichtlich bestimmter Märkte und Anwendungen bei bestimmten Behörden eingereicht. Wenden Sie sich an Ihre Siemens-Vertretung, wenn Sie eine Liste mit den aktuellen Zulassungen für die einzelnen Bestellnummern benötigen.

Klassifizierungsgesellschaften:

- ABS (American Bureau of Shipping)
- BV (Bureau Veritas)
- DNV (Det Norske Veritas)
- GL (Germanischer Lloyd)
- LRS (Lloyds Register of Shipping)
- Class NK (Nippon Kaiji Kyokai)

### Industrieumgebungen

Das Automatisierungssystem S7-1200 wurde für den Einsatz in Industrieumgebungen entwickelt.

Tabelle A- 1 Industrieumgebungen

Anwendungsgebiet	Anforderungen an die Störaussendung	Anforderungen an die Störfestigkeit
Industrie	EN 61000-6-4:2007	EN 61000-6-2:2005

### Elektromagnetische Verträglichkeit (EMV)

Die elektromagnetische Verträglichkeit (EMV) eines elektrischen Geräts ist dessen Fähigkeit, in einer elektromagnetischen Umgebung bestimmungsgemäß zu funktionieren und keine elektromagnetischen Störungen auszusenden, die den Betrieb anderer elektrischer Geräte in der Umgebung beeinträchtigen könnten.

Tabelle A- 2 Störfestigkeit EN 61000-6-2

Elektromagnetische Verträglichkeit - Entstörung nach EN 61000-6-2	
EN 61000-4-2 Elektrostatistische Entladung	8 kV Entladung durch die Luft an allen Oberflächen 6 kV Entladung durch Kontakt mit freiliegenden leitenden Oberflächen
EN 61000-4-3 Prüfung der Störfestigkeit gegen hochfrequente elektromagnetische Felder	80 bis 1000 MHz, 10 V/m, 80 % AM bei 1 kHz 1,4 bis 2,0 GHz, 3 V/m, 80 % AM bei 1 kHz 2,0 bis 2,7 GHz, 1 V/m, 80 % AM bei 1 kHz
EN 61000-4-4 Schnelle transiente Störgröße	2 kV, 5 kHz bei Kopplungsnetz zu AC und DC Systemspannung 2 kV, 5 kHz bei Kopplungsklemme zu Ein/Ausgängen
EN 61000-4-5 Stoßwellenfestigkeit	AC-Systeme - 2 kV Gleichtakt, 1kV Gegentakt DC-Systeme - 2 kV Gleichtakt, 1kV Gegentakt Für DC-Systeme (E/A-Signale, DC-Stromversorgungen) ist ein externer Schutz erforderlich.
EN 61000-4-6 Leitungsgeführte Störungen	150 kHz bis 80 MHz, 10 V effektiv, 80% AM bei 1 kHz
EN 61000-4-11 Spannungseinbrüche	AC-Systeme 0% für 1 Zyklus, 40% für 12 Zyklen und 70% für 30 Zyklen bei 60 Hz

Tabelle A- 3 Leitungsgeführte und abgestrahlte Störaussendungen nach EN 61000-6-4

Elektromagnetische Verträglichkeit - Leitungsgeführte und abgestrahlte Störaussendungen nach EN 61000-6-4		
Leitungsgeführte Störaussendungen EN 55011, Klasse A, Gruppe 1	0,15 MHz bis 0,5 MHz	<79 dB (µV) Quasi-Spitze; <66 dB (µV) Mittelwert
	0,5 MHz bis 5 MHz	<73 dB (µV) Quasi-Spitze; <60 dB (µV) Mittelwert
	5 MHz bis 30 MHz	<73 dB (µV) Quasi-Spitze; <60 dB (µV) Mittelwert
Abgestrahlte Störaussendungen EN 55011, Klasse A, Gruppe 1	30 MHz bis 230 MHz	<40 dB (µV/m) Quasi-Spitze; gemessen in einer Entfernung von 10 m
	230 MHz bis 1 GHz	<47 dB (µV/m) Quasi-Spitze; gemessen in einer Entfernung von 10 m

### Umgebungsbedingungen

Tabelle A- 4 Transport und Lagerung

Umgebungsbedingungen - Transport und Lagerung	
EN 6006822, Test Bb, trockene Wärme und EN 6006821 Test Ab, Kälte	-40 °C bis +70 °C
EN 60068230, Test Db, feuchte Wärme	25° C bis 55° C, 95% Luftfeuchtigkeit
EN 60068-2-14, Test Na, Temperaturschock	-40 °C bis +70 °C, Haltezeit 3 Stunden, 5 Zyklen

Umgebungsbedingungen - Transport und Lagerung	
EN 60068-2-32 Freier Fall	0,3 m, 5 Mal, in Versandverpackung
Atmosphärischer Druck	1080 bis 660 hPa (entspricht einer Höhe von -1000 bis 3500 m)

Tabelle A- 5 Betriebsbedingungen

Umgebungsbedingungen - Betrieb	
Umgebungstemperaturen (Luftzufuhr 25 mm unterhalb des Geräts)	-20 °C bis 60 °C horizontale Montage -20 °C bis 50 °C vertikale Montage 95% Luftfeuchtigkeit, nicht kondensierend Sofern nicht anders angegeben
Atmosphärischer Druck	1080 bis 795 hPa (entspricht einer Höhe von -1000 bis 2000m)
Konzentration von Schmutzstoffen	SO <sub>2</sub> : < 0,5 ppm; H <sub>2</sub> S: < 0,1 ppm; rel. LF < 60% nicht kondensierend
EN 60068-2-14, Test Nb, Temperaturveränderung	5° C bis 55° C, 3° C/Minute
EN 60068227 Mechanische Stoßbeanspruchung	15 G, Impuls 11 ms, 6 Stöße auf jeder der 3 Achsen
EN 6006826 Sinusschwingung	Hutschienenmontage: 3,5 mm von 5-9 Hz, 1 G von 9 - 150 Hz Schalttafeleinbau: 7,0 mm von 5 bis 9 Hz; 2 G von 9 bis 150 Hz 10 Ablenkungen je Achse, 1 Oktave/Minute

**ACHTUNG**

Bei Systemen, die im Bereich von -20 °C bis 0 °C anlaufen müssen, muss das Anwenderprogramm die Aktivierung der Ausgänge nach dem Anlauf 10 Sekunden lang verzögern.

Tabelle A- 6 Hochspannungsisolationsprüfung

Hochspannungsisolationsprüfung	
Stromkreis mit 24V/5VNennspannung	520 V DC (Typprüfung der optischen Potentialtrennungsgrenzen)
115/230VStromkreis an Erde	1500 V AC
115/230VStromkreis an 115/230VStromkreis	1500 V AC
115/230VStromkreis an 24/5VStromkreis	1500 V AC (3000 V AC / 4242 V DC Typprüfung)
Ethernet-Port an 24-V/5-V-Stromkreise und Erde <sup>1</sup>	1500 V AC (nur Typprüfung)

<sup>1</sup> Die Potentialtrennung des Ethernet-Ports ist dafür ausgelegt, die Gefahr während kurzfristiger Netzausfälle von gefährlichen Spannungen zu begrenzen. Sie entspricht nicht den Sicherheitsanforderungen der routinemäßigen Potentialtrennung von AC-Netzspannungen.

**Schutzart**

- Schutzklasse II nach EN 61131-2 (Schutzleiter nicht erforderlich)

### Schutzgrad

- IP20 Mechanischer Schutz, EN 60529
- Schutz gegen direkte Berührung von Hochspannung wie mit genormter Sonde ermittelt. Externer Schutz erforderlich gegen Staub, Schmutz, Wasser und Fremdkörper mit einem Durchmesser von < 12,5 mm.

### Bemessungsspannungen

Tabelle A- 7 Bemessungsspannungen

Nennspannung	Toleranz
24 V DC	20,4 V DC bis 28,8 V DC 22,0 V DC bis 28,8 V DC bei Umgebungstemperatur -20 °C bis 0 °C
120/230 V AC	85 bis 264 V AC, 47 bis 63 Hz

<b>ACHTUNG</b>
Wenn ein mechanischer Kontakt die Ausgangsspannung zur S7-1200 CPU oder einem digitalen Signalmodul einschaltet, wird ca. 50 Mikrosekunden lang das Signal "1" an die Digitalausgänge gesendet. Dies kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann. Dies müssen Sie berücksichtigen, vor allem, wenn Sie mit Geräten arbeiten, die auf kurze Impulse reagieren.

### Verpolschutz

Verpolschutz ist vorhanden bei allen Klemmenpaaren mit +24-V-DC-Spannungsversorgung oder anwenderseitiger Eingangsspannung für CPUs, Signalmodule (SMs) und Signalboards (SBs). Trotzdem sind Beschädigungen des System weiterhin dadurch möglich, dass unterschiedliche Klemmenpaare mit entgegengesetzter Polarität verdrahtet werden.

Einige der 24-V-DC-Eingangsports des S7-1200 Systems sind miteinander verbunden, wobei ein logischer Bezugsleiter mehrere M-Klemmen verbindet. Beispielsweise sind die folgenden Stromkreise miteinander verbunden, sofern sie in den Datenblättern als "nicht potentialgetrennt" angegeben sind: die 24-V-DC-Versorgung der CPU, der Leistungseingang für die Relaispule eines SM oder die Versorgung eines nicht potentialgetrennten Analogeingangs. Alle nicht potentialgetrennten M-Klemmen müssen an dasselbe externe Bezugspotential angeschlossen werden.

### WARNUNG

Wenn Sie nicht potentialgetrennte M-Klemmen an verschiedene Bezugspotentiale anschließen, verursacht dies unbeabsichtigten Stromfluss, der zu Beschädigung oder unvorhersehbarem Betriebsverhalten des Zielsystems und angeschlossener Geräte führen kann.

Die Nichteinhaltung dieser Richtlinien kann Schaden oder unvorhersehbares Betriebsverhalten verursachen, was zu Tod oder schwerer Körperverletzung und/oder Sachschaden führen kann.

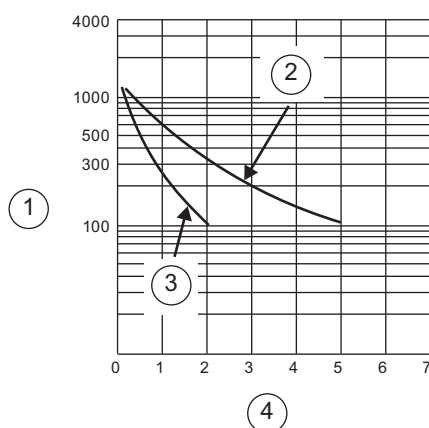
Schließen Sie stets alle nicht potentialgetrennten M-Klemmen in einem S7-1200 System an dasselbe Bezugspotential an.

## DC-Ausgänge

Es stehen keine kurzschlussfesten Schaltungen für die Gleichspannungsausgänge an CPUs, Signalmodulen (SMs) und Signalboards (SBs) zur Verfügung.

## Lebensdauer eines Relais

Die typischen Leistungsdaten, die von Relais-Herstellern zur Verfügung gestellt werden, sind nachstehend aufgeführt. Die tatsächliche Leistungsfähigkeit richtet sich nach der jeweiligen Verwendung. Ein externer Schutzkreis, der der Last angepasst ist, verlängert die Lebensdauer der Kontakte.



- ① Lebensdauer (x 10<sup>3</sup> Operationen)
- ② Widerstandslast 250 V AC  
Widerstandslast 30 V DC
- ③ Induktive Last 250 V AC (p.f.=0,4)  
Induktive Last 30 V DC (L/R=7ms)
- ④ Nennbetriebsstrom (A)

## A.2 CPU 1211C

### A.2.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A- 8 Allgemeine technische Daten

Technische Daten	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Bestellnummer	6ES7 211-1BE31-0XB0	6ES7 211-1HE31-0XB0	6ES7 211-1AE31-0XB0
Abmessungen B x H x T (mm)	90 x 100 x 75	90 x 100 x 75	90 x 100 x 75
Versandgewicht	420 Gramm	380 Gramm	370 Gramm
Leistungsverlust	10 W	8 W	8 W
Verfügbarer Strom (CM-Bus)	max. 750 mA (5 V DC)	max. 750 mA (5 V DC)	max. 750 mA (5 V DC)
Verfügbarer Strom (24 V DC)	max. 300 mA (Geberversorgung)	max. 300 mA (Geberversorgung)	max. 300 mA (Geberversorgung)
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang	4 mA/Eingang	4 mA/Eingang

Tabelle A- 9 CPU-Merkmale

Technische Daten	Beschreibung	
Anwenderspeicher <sup>1</sup>	Arbeitsspeicher	30 KB
	Last	1 MB, intern, erweiterbar bis auf SD-Kartengröße
	Remanent	10 KB
Integrierte digitale E/A	6 Eingänge/4 Ausgänge	
Integrierte analoge E/A	2 Eingänge	
Größe des Prozessabbilds	1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)	
Merker (M)	4096 Byte	
Temporärer (lokaler) Speicher	<ul style="list-style-type: none"> <li>• 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs)</li> <li>• 4 KB für Standardalarmereignisse, einschließlich FBs und FCs</li> <li>• 4 KB für Fehleralarmereignisse, einschließlich FBs und FCs</li> </ul>	
Zusätzliche Signalmodule	Keine	
Erweiterung SB, CB, BB	max. 1	
Zusätzliche Kommunikationsmodule	max. 3 CM	
Schnelle Zähler	3 integrierte E/A, 5 mit Signalboard, siehe Tabelle HSC-Eingangszuweisungen bei der CPU 1211C (Seite 359) <ul style="list-style-type: none"> <li>• Einphasenzähler: 3 bei 100 kHz, SB: 2 bei 30 kHz</li> <li>• A/B-Zähler: 3 bei 80 kHz, SB: 2 bei 20 kHz</li> </ul>	
Impulsausgänge <sup>2</sup>	4	
Eingänge für Impulsabgriff	6	

Technische Daten	Beschreibung
Verzögerungs-/Weckalarme	4 mit Auflösung von 1 ms
Flankenalarme	6 steigend und 6 fallend (10 und 10 mit optionalem Signalboard)
Memory Card	SIMATIC Memory Card (optional)
Genauigkeit Echtzeituhr	+/- 60 Sekunden/Monat
Pufferung Echtzeituhr	Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

- Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU begrenzt. Die Anzahl der unterstützten OBs, FCs, FBs und DBs und die Größe der einzelnen Bausteine ist nicht begrenzt. Die einzige Begrenzung ist die Gesamtspeicherkapazität.
- Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Tabelle A- 10 Leistung

Art der Anweisung	Ausführungsgeschwindigkeit
Boolesch	0,08 µs/Operation
Wort übertragen	1,7 µs/Operation
Realzahlenarithmetik	2,3 µs/Operation

## A.2.2 Von der CPU 1211C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A- 11 Von der CPU 1211C unterstützte Bausteine, Zeiten und Zähler

Element	Beschreibung	
Bausteine	Typ	OB, FB, FC, DB
	Größe	30 KB
	Anzahl	Bis 1024 Bausteine insgesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	1 bis 65535 (z. B. FB 1 bis FB 65535)
	Schachteltiefe	16 aus Zyklus- oder Anlauf-OBs, 4 aus Verzögerungsalarm-, Uhrzeitalarm-, Weckalarm-, Prozessalarm-, Zeitfehler- oder Diagnosefehler-OBs
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.
OBs	Programmzyklus	Mehrere: OB 1, OB 200 bis OB 65535
	Anlauf	Mehrere: OB 100, OB 200 bis OB 65535
	Verzögerungsalarme und Weckalarme	4 <sup>1</sup> (1 pro Ereignis): OB 200 bis OB 65535
	Prozessalarme (Flanken und HSC)	50 (1 pro Ereignis): OB 200 bis OB 65535
	Zeitfehleralarme	1: OB 80
	Diagnosefehleralarme	1: OB 82

A.2 CPU 1211C

Element		Beschreibung
Zeiten	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Byte pro Zeit
Zähler	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"> <li>• SInt, USInt: 3 Byte</li> <li>• Int, UInt: 6 Byte</li> <li>• DInt, UDInt: 12 Byte</li> </ul>

<sup>1</sup> Verzögerungs- und Weckalarme nutzen dieselben Ressourcen in der CPU. Es darf insgesamt maximal 4 dieser Alarme geben (Verzögerungs- plus Weckalarme). 4 Verzögerungsalarme und 4 Weckalarme sind nicht möglich.

Tabelle A- 12 Kommunikation

Technische Daten	Beschreibung
Schnittstellen	1
Typ	Ethernet
HMI-Gerät <sup>1</sup>	3
Programmiergerät (PG)	1
Anschlüsse	<ul style="list-style-type: none"> <li>• 8 für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV</li> <li>• 3 für die S7-Kommunikation über Server-GET/PUT (CPU-zu-CPU)</li> <li>• 8 für die S7-Kommunikation über Client-GET/PUT (CPU-zu-CPU)</li> </ul>
Datenraten	10/100 MBit/s
Elektrische Trennung (externes Signal zu PLC-Logik)	Wandler potentialgetrennt, 1500 V AC, nur für kurzfristige Sicherheit
Kabelart	CAT5e geschirmt

<sup>1</sup> Die CPU bietet zweckbestimmte HMI-Verbindungen, um bis zu 3 HMI-Geräte zu unterstützen. (Sie können bis zu 2 SIMATIC Comfort Panels haben.) Wie viele HMI-Geräte insgesamt unterstützt werden, hängt von den Typen der HMI-Panels in Ihrer Konfiguration ab. Sie können beispielsweise bis zu drei SIMATIC Basic Panels an Ihre CPU anschließen, oder Sie können bis zu zwei SIMATIC Comfort Panels und ein zusätzliches Basic Panel anschließen.

Tabelle A- 13 Spannungsversorgung

Technische Daten	CPU 1211C	CPU 1211C	CPU 1211C
	AC/DC/Relais	DC/DC/Relais	DC/DC/DC
Spannungsbereich	85 bis 264 V AC	20,4 V DC bis 28,8 V DC 22,0 VDC bis 28,8 V DC bei Umgebungstemperatur -20 °C bis 0 °C	
Netzfrequenz	47 bis 63 Hz	--	--
Eingangsstrom nur CPU bei max. Last	60 mA bei 120 V AC 30 mA bei 240 V AC	300 mA bei 24 V DC	300 mA bei 24 V DC



Technische Daten	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
CPU mit allen Erweiterungsbaugruppen bei max. Last	180 mA bei 120 V AC 90 mA bei 240 V AC	900 mA bei 24 V DC	900 mA bei 24 V DC
Einschaltstrom (max.)	20 A bei 264 V AC	12 A bei 28,8 V DC	12 A bei 28,8 V DC
Elektrische Trennung (Eingangsleistung zu Logik)	1500 V AC	Nicht elektrisch getrennt	Nicht elektrisch getrennt
Kriechstrom an Erde, AC-Leitung an Funktionserde	max. 0,5 mA	--	--
Verzögerungszeit (Spannungsverlust)	20 ms bei 120 V AC 80 ms bei 240 V AC	10 ms bei 24 V DC	10 ms bei 24 V DC
Interne Sicherung, nicht durch Anwender austauschbar	3 A, 250 V, träge	3 A, 250 V, träge	3 A, 250 V, träge

Tabelle A- 14 Geberversorgung

Technische Daten	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Spannungsbereich	20,4 bis 28,8 V DC	L + minus 4 V DC min. L+ minus 5 V DC min. bei einer Umgebungstemperatur von -20 °C bis 0 °C	
Nennausgangsstrom (max.)	300 mA (kurzschlussfest)	300 mA (kurzschlussfest)	300 mA (kurzschlussfest)
Max. Welligkeit/Störströme (<10 MHz)	< 1 V Spitze-Spitze	Wie Eingangsleitung	Wie Eingangsleitung
Elektrische Trennung (CPU-Logik zu Sensorversorgung)	Nicht elektrisch getrennt	Nicht elektrisch getrennt	Nicht elektrisch getrennt

### A.2.3 Digitale Eingänge und Ausgänge

Tabelle A- 15 Digitaleingänge

Technische Daten	CPU 1211C AC/DC/Relais, CPU 1211C DC/DC/Relais und CPU 1211C DC/DC/DC
Anzahl der Eingänge	6
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)
Nennspannung	24 V DC bei 4 mA, Nennwert
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute
Potenzialgetrennte Gruppen	1

Technische Daten	CPU 1211C AC/DC/Relais, CPU 1211C DC/DC/Relais und CPU 1211C DC/DC/DC
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)
HSC Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	Einphasenzähler: 100 kHz A/B-Zähler: 80 kHz
Anzahl gleichzeitig eingeschalteter Eingänge	6 bei 60 °C horizontal oder 50 °C vertikal
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge

Tabelle A- 16 Digitale Ausgänge

Technische Daten	CPU 1211C AC/DC/Relais und CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Ausgänge	4	4
Typ	Relais, Trockenkontakt	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule zu Kontakt) Keine (Spule zu Logik)	500 V AC für 1 Minute
Isolationswiderstand	min. 100 MΩ, wenn neu	--
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	--
Potentialgetrennte Gruppen	1	1
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Maximale Schaltfrequenz Relais	1 Hz	--
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 µs von Aus nach Ein max. 3,0 µs von Ein nach Aus
Frequenz Impulsgenerator (Aa.0 und Aa.2)	Nicht empfehlenswert <sup>1</sup>	max. 100 kHz, min. 2 Hz <sup>2</sup>
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)

Technische Daten	CPU 1211C AC/DC/Relais und CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Anzahl gleichzeitig eingeschalteter Ausgänge	4 bei 60 °C horizontal oder 50 °C vertikal	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt

- <sup>1</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.
- <sup>2</sup> Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

## A.2.4 Analoge Eingänge

Tabelle A- 17 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27.649 bis 32.511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32.512 bis 32.767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 756).
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25°C / -20 bis 60°C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrehtes Leiterpaar

### A.2.4.1 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A- 18 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
<b>Abtastzeit</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

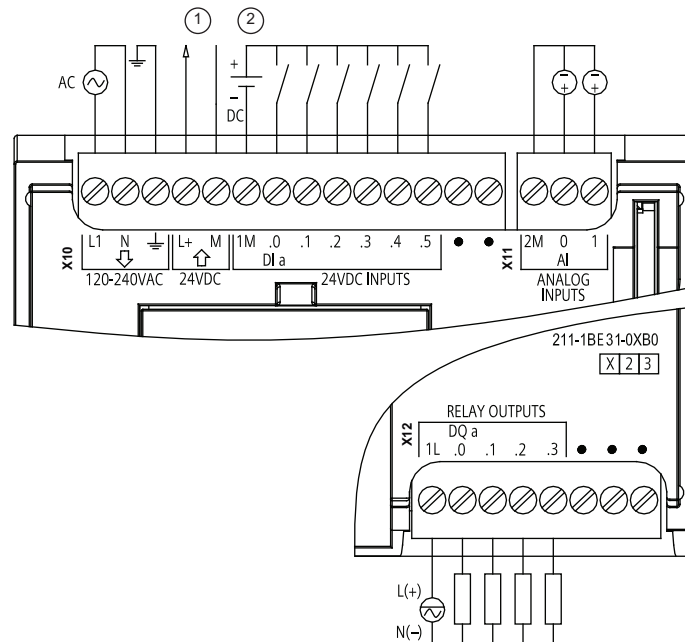
### A.2.4.2 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A- 19 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

## A.2.5 Schaltpläne der CPU 1211C

Tabelle A- 20 CPU 1211C AC/DC/Relais (6ES7 211-1BE31-0XB0)



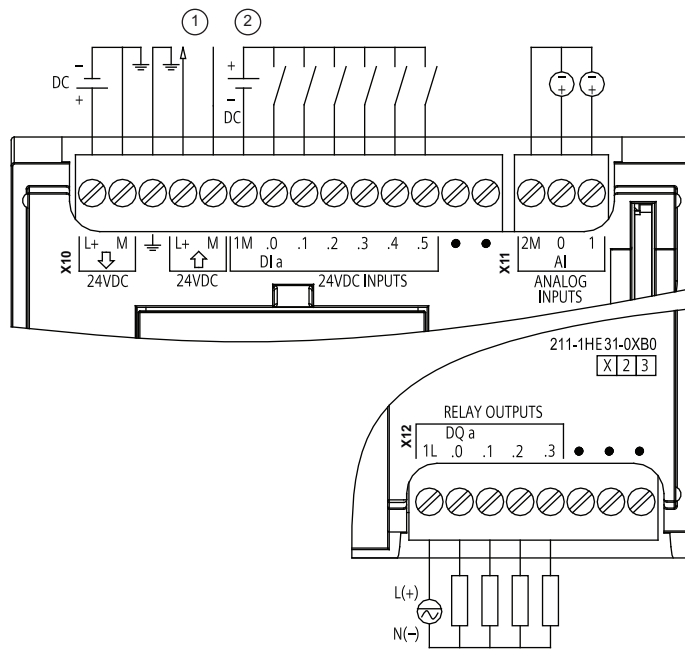
- ① 24-V-DC-  
Geberverspannung  
Um zusätzliche  
Störfestigkeit zu  
erreichen, schließen  
Sie "M" an Masse an,  
auch wenn Sie keine  
Geberversorgung  
verwenden.
- ② Bei stromziehenden  
Eingängen "-" an "M"  
anschließen  
(abgebildet).  
Bei stromliefernden  
Eingängen "+" an "M"  
anschließen.

Hinweis: X11-  
Steckverbinder müssen  
vergoldet sein. Im Anhang  
C, Ersatzteile, finden Sie  
die Bestellnummer.

Tabelle A- 21 Anschlussbelegung für die CPU 1211C AC/DC/Relais (6ES7 211-1BE31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / 24-V-DC- Sensorausgang	--	DO a.2
5	M / 24-V-DC- Sensorausgang	--	DO a.3
6	1M	--	Kein Anschluss
7	DI a.0	--	Kein Anschluss
8	DI a.1	--	Kein Anschluss
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Kein Anschluss	--	--
14	Kein Anschluss	--	--

Tabelle A- 22 CPU 1211C DC/DC/Relais (6ES7 211-1HE31-0XB0)



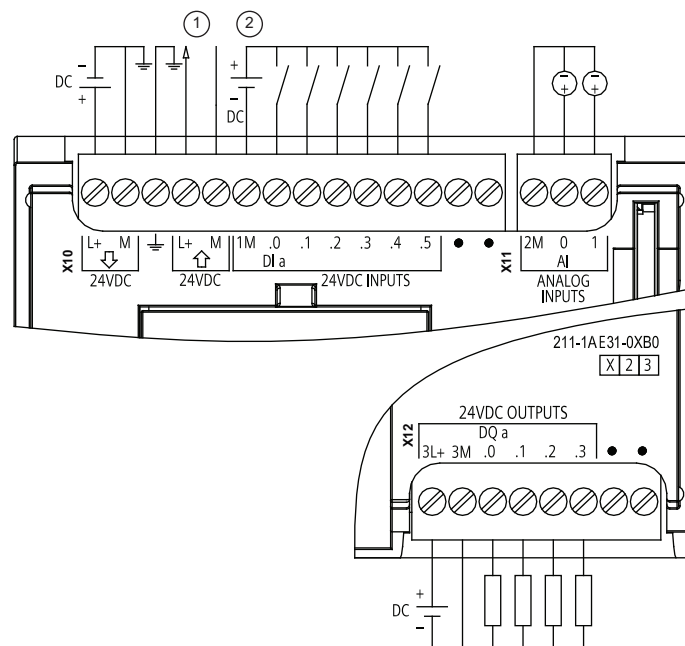
- ① 24-V-DC-Geberspannung  
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Gebersversorgung verwenden.
- ② Bei stromziehenden Eingängen "-" an "M" anschließen (gezeigt). Bei stromliefernden Eingängen "+" an "M" anschließen.

Hinweis: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 23 Anschlussbelegung für die CPU 1211C DC/DC/Relais (6ES7 211-1HE31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / 24-V-DC-Sensorausgang	--	DO a.2
5	M / 24-V-DC-Sensorausgang	--	DO a.3
6	1M	--	Kein Anschluss
7	DI a.0	--	Kein Anschluss
8	DI a.1	--	Kein Anschluss
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Kein Anschluss	--	--
14	Kein Anschluss	--	--

Tabelle A- 24 CPU 1211C DC/DC/DC (6ES7 211-1AE31-0XB0)



- ① 24-V-DC-  
Gegenspannung  
Um zusätzliche  
Störfestigkeit zu  
erreichen, schließen  
Sie "M" an Masse an,  
auch wenn Sie keine  
Gegenspannung  
verwenden.
- ② Bei stromziehenden  
Eingängen "-" an "M"  
anschließen (gezeigt).  
Bei stromliefernden  
Eingängen "+" an "M"  
anschließen.

Hinweis: X11-  
Steckverbinder müssen  
vergoldet sein. Im Anhang  
C, Ersatzteile, finden Sie  
die Bestellnummer.

Tabelle A- 25 Anschlussbelegung für die CPU 1211C DC/DC/DC (6ES7 211-1AE31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Funktionserde	AI 1	DO a.0
4	L+ / 24-V-DC- Sensorausgang	--	DO a.1
5	M / 24-V-DC- Sensorausgang	--	DO a.2
6	1M	--	DO a.3
7	DI a.0	--	Kein Anschluss
8	DI a.1	--	Kein Anschluss
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Kein Anschluss	--	--
14	Kein Anschluss	--	--

**Hinweis**

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

## A.3 CPU 1212C

### A.3.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A- 26 Allgemein

Technische Daten	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Bestellnummer	6ES7 212-1BE31-0XB0	6ES7 212-1HE31-0XB0	6ES7 212-1AE31-0XB0
Abmessungen B x H x T (mm)	90 x 100 x 75	90 x 100 x 75	90 x 100 x 75
Versandgewicht	425 Gramm	385 Gramm	370 Gramm
Leistungsverlust	11 W	9 W	9 W
Verfügbare Strom (SM- und CM-Bus)	max. 1000 mA (5 V DC)	max. 1000 mA (5 V DC)	max. 1000 mA (5 V DC)
Verfügbare Strom (24 V DC)	max. 300 mA (Geberversorgung)	max. 300 mA (Geberversorgung)	max. 300 mA (Geberversorgung)
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang	4 mA/Eingang	4 mA/Eingang

Tabelle A- 27 CPU-Merkmale

Technische Daten	Beschreibung	
Anwenderspeich er <sup>1</sup>	Arbeitsspeicher	50 KB
	Ladespeicher	1 MB, intern, erweiterbar bis auf SD-Kartengröße
	Remanent	10 KB
Integrierte digitale E/A	8 Eingänge/6 Ausgänge	
Integrierte analoge E/A	2 Eingänge	
Größe des Prozessabbilds	1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)	
Merker (M)	4096 Byte	
Temporärer (lokaler) Speicher	<ul style="list-style-type: none"> <li>• 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs)</li> <li>• 4 KB für Standardalarmereignisse, einschließlich FBs und FCs</li> <li>• 4 KB für Fehleralarmereignisse, einschließlich FBs und FCs</li> </ul>	
Zusätzliche Signalmodule	max. 2 SMs	
Erweiterung SB, CB, BB	max. 1	



Technische Daten	Beschreibung
Zusätzliche Kommunikationsmodule	max. 3 CM
Schnelle Zähler	5 integrierte E/A, 6 mit Signalboard, siehe Tabelle HSC-Eingangszuweisungen bei der CPU 1212C (Seite 359) <ul style="list-style-type: none"> <li>• Einphasenzähler: 3 bei 100 kHz und 1 bei 30 kHz Takt, SB: 2 bei 30 kHz</li> <li>• A/B-Zähler: 3 bei 80 kHz und 1 bei 20 kHz Takt, SB: 2 bei 20 kHz</li> </ul>
Impulsausgänge <sup>2</sup>	4
Eingänge für Impulsabgriff	8
Verzögerungs-/Weckalarne	4 mit Auflösung von 1 ms
Flankenalarne	8 steigend und 8 fallend (12 und 12 mit optionalem Signalboard)
Memory Card	SIMATIC Memory Card (optional)
Genauigkeit Echtzeituhr	+/- 60 Sekunden/Monat
Pufferung Echtzeituhr	Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

<sup>1</sup> Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU begrenzt. Die Anzahl der unterstützten OBs, FCs, FBs und DBs und die Größe der einzelnen Bausteine ist nicht begrenzt. Die einzige Begrenzung ist die Gesamtspeicherkapazität.

<sup>2</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Tabelle A- 28 Leistung

Art der Anweisung	Ausführungsgeschwindigkeit
Boolesch	0,08 µs/Operation
Wort übertragen	1,7 µs/Operation
Realzahlenarithmetik	2,3 µs/Operation

### A.3.2 Von der CPU 1212C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A- 29 Von der CPU 1212C unterstützte Bausteine, Zeiten und Zähler

Element	Beschreibung	
Bausteine	Typ	OB, FB, FC, DB
	Größe	50 KB
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	1 bis 65535 (z. B. FB 1 bis FB 65535)
	Schachtelungstiefe	16 aus Zyklus- oder Anlauf-OBs, 4 aus Verzögerungsalarm-, Uhrzeitalarm-, Weckalarm-, Prozessalarm-, Zeitfehler- oder Diagnosefehler-OBs
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.
OBs	Programmzyklus	Mehrere: OB 1, OB 200 bis OB 65535
	Anlauf	Mehrere: OB 100, OB 200 bis OB 65535



Tabelle A- 31 Spannungsversorgung

Technische Daten		CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Spannungsbereich		85 bis 264 V AC	20,4 V DC bis 28,8 V DC 22,0 VDC bis 28,8 V DC bei Umgebungstemperatur -20 °C bis 0 °C	
Netzfrequenz		47 bis 63 Hz	--	--
Einschaltstrom (max. Last)	Nur CPU	80 mA bei 120 V AC 40 mA bei 240 V AC	400 mA bei 24 V DC	400 mA bei 24 V DC
	CPU mit allen Erweiterungsbaugruppen	240 mA bei 120 V AC 120 mA bei 240 V AC	1200 mA bei 24 V DC	1200 mA bei 24 V DC
Einschaltstrom (max.)		20 A bei 264 V AC	12 A bei 28,8 V DC	12 A bei 28,8 V DC
Elektrische Trennung (Eingangsleistung zu Logik)		1500 V AC	Nicht elektrisch getrennt	Nicht elektrisch getrennt
Kriechstrom an Erde, AC-Leitung an Funktionserde		max. 0,5 mA	--	--
Verzögerungszeit (Spannungsverlust)		20 ms bei 120 V AC 80 ms bei 240 V AC	10 ms bei 24 V DC	10 ms bei 24 V DC
Interne Sicherung, nicht durch Anwender austauschbar		3 A, 250 V, träge	3 A, 250 V, träge	3 A, 250 V, träge

Tabelle A- 32 Geberversorgung

Technische Daten		CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Spannungsbereich		20,4 bis 28,8 V DC	L + minus 4 V DC min. L+ minus 5 V DC min. bei einer Umgebungstemperatur von -20 °C bis 0 °C	
Nennausgangsstrom (max.)		300 mA (kurzschlussfest)	300 mA (kurzschlussfest)	300 mA (kurzschlussfest)
Max. Welligkeit/Störströme (<10 MHz)		< 1 V Spitze-Spitze	Wie Eingangsleitung	Wie Eingangsleitung
Elektrische Trennung (CPU-Logik zu Sensorversorgung)		Nicht elektrisch getrennt	Nicht elektrisch getrennt	Nicht elektrisch getrennt

### A.3.3 Digitale Eingänge und Ausgänge

Tabelle A- 33 Digitaleingänge

Technische Daten	CPU 1212C AC/DC/Relais, DC/DC/Relais und DC/DC/DC
Anzahl der Eingänge	8
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)
Nennspannung	24 V DC bei 4 mA, Nennwert
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)
HSC Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	Einphasenzähler: 100 kHz (Ea.0 bis Ea.5) und 30 kHz (Ea.6 bis Ea.7) A/B-Zähler: 80 kHz (Ea.0 bis Ea.5) und 20 kHz (Ea.6 bis Ea.7)
Anzahl gleichzeitig eingeschalteter Eingänge	8 bei 60 °C horizontal, 50 °C vertikal
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge

Tabelle A- 34 Digitale Ausgänge

Technische Daten	CPU 1212C AC/DC/Relais und DC/DC/Relais	CPU 1212C DC/DC/DC
Ausgänge	6	6
Typ	Relais, Trockenkontakt	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule zu Kontakt) Keine (Spule zu Logik)	500 V AC für 1 Minute
Isolationswiderstand	min. 100 MΩ, wenn neu	--
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	--
Potentialgetrennte Gruppen	2	1

Technische Daten	CPU 1212C AC/DC/Relais und DC/DC/Relais	CPU 1212C DC/DC/DC
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 µs von Aus nach Ein max. 3,0 µs von Ein nach Aus
Schaltverzögerung (Aa.4 bis Aa.5)	max. 10 ms	max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	--
Frequenz Impulsgenerator (Aa.0 und Aa.2)	Nicht empfehlenswert <sup>1</sup>	max. 100 kHz, min. 2 Hz <sup>2</sup>
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Anzahl gleichzeitig eingeschalteter Ausgänge	6 bis 60 °C horizontal, 50 °C vertikal	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt

<sup>1</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

<sup>2</sup> Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

### A.3.4 Analoge Eingänge

Tabelle A- 35 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27.649 bis 32.511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32.512 bis 32.767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 766).

Technische Daten	Beschreibung
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25°C / -20 bis 60°C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrehtes Leiterpaar

### A.3.4.1 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A- 36 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
<b>Abtastzeit</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

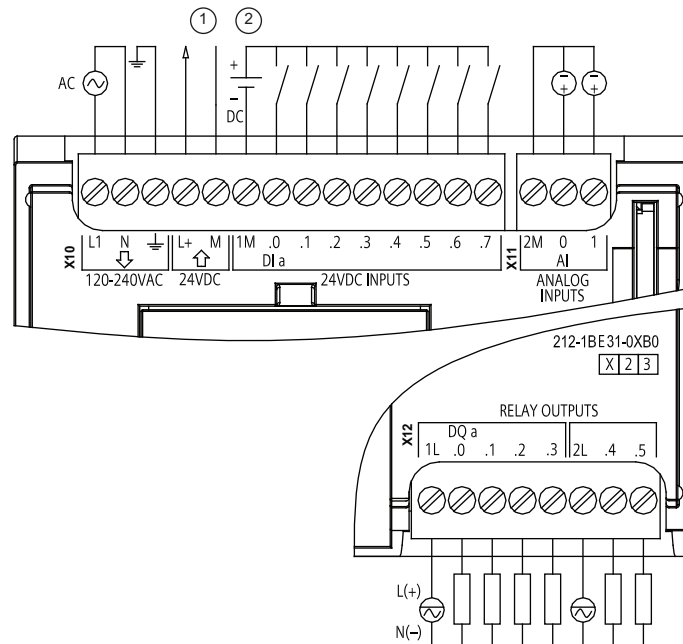
### A.3.4.2 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A- 37 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

### A.3.5 Schaltpläne der CPU 1212C

Tabelle A- 38 CPU 1212C AC/DC/Relais (6ES7 212-1BE31-0XB0)



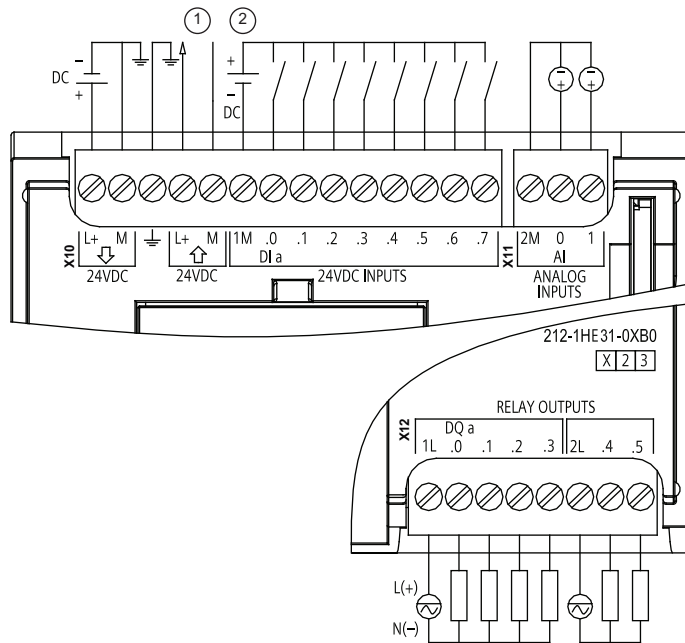
- ① 24-V-DC-  
Geberspannung  
Um zusätzliche  
Störfestigkeit zu  
erreichen, schließen  
Sie "M" an Masse an,  
auch wenn Sie keine  
Gebersversorgung  
verwenden.
- ② Bei stromziehenden  
Eingängen "-" an "M"  
anschießen  
(abgebildet).  
Bei stromliefernden  
Eingängen "+" an "M"  
anschießen.

Hinweis: X11-  
Steckverbinder müssen  
vergoldet sein. Im Anhang  
C, Ersatzteile, finden Sie  
die Bestellnummer.

Tabelle A- 39 Anschlussbelegung für die CPU 1212C AC/DC/Relais (6ES7 212-1BE31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / 24-V-DC- Sensorausgang	--	DO a.2
5	M / 24-V-DC- Sensorausgang	--	DO a.3
6	1M	--	2L
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tabelle A- 40 CPU 1212C DC/DC/Relais (6ES7 212-1HE31-0XB0)



- ① 24-V-DC-Geberspannung  
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Gebersversorgung verwenden.
- ② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).  
Bei stromliefernden Eingängen "+" an "M" anschließen.

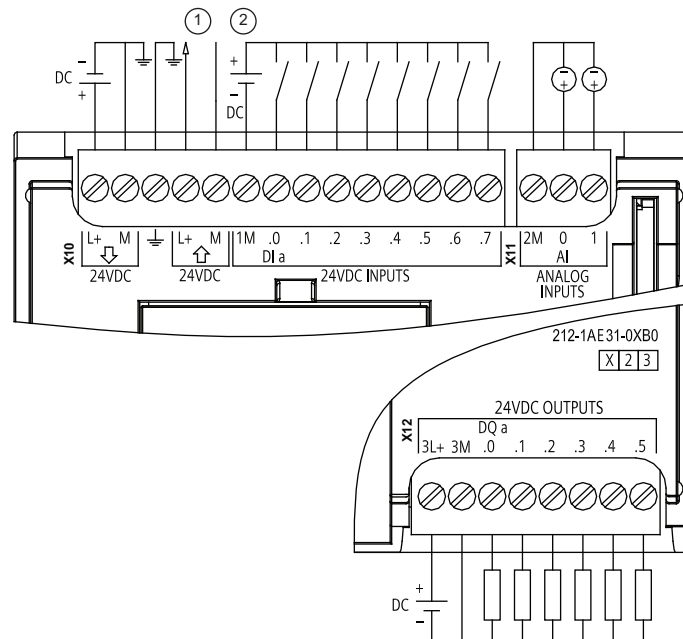
Hinweis: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 41 Anschlussbelegung für die CPU 1212C DC/DC/Relais (6ES7 212-1HE31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / 24-V-DC-Sensorausgang	--	DO a.2
5	M / 24-V-DC-Sensorausgang	--	DO a.3
6	1M	--	2L
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--



Tabelle A- 42 CPU 1212C DC/DC/DC (6ES7 -212-1AE31-0XB0)



- ① 24-V-DC-  
Geberspannung  
Um zusätzliche  
Störfestigkeit zu  
erreichen, schließen  
Sie "M" an Masse an,  
auch wenn Sie keine  
Geberversorgung  
verwenden.
- ② Bei stromziehenden  
Eingängen "-" an "M"  
anschließen  
(abgebildet).  
Bei stromliefernden  
Eingängen "+" an "M"  
anschließen.

Hinweis: X11-  
Steckverbinder müssen  
vergoldet sein. Im Anhang  
C, Ersatzteile, finden Sie  
die Bestellnummer.

Tabelle A- 43 Anschlussbelegung für die CPU 1212C DC/DC/DC (6ES7 212-1AE31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Funktionserde	AI 1	DO a.0
4	L+ / 24-V-DC- Sensorausgang	--	DO a.1
5	M / 24-V-DC- Sensorausgang	--	DO a.2
6	1M	--	DO a.3
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

**Hinweis**

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

## A.4 CPU 1214C

### A.4.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A- 44 Allgemein

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Bestellnummer	6ES7 214-1BG31-0XB0	6ES7 214-1HG31-0XB0	6ES7 214-1AG31-0XB0
Abmessungen (B x H x T) (mm)	110 x 100 x 75	110 x 100 x 75	110 x 100 x 75
Versandgewicht	475 Gramm	435 Gramm	415 Gramm
Leistungsverlust	14 W	12 W	12 W
Verfügbarer Strom (SM- und CM-Bus)	max. 1600 mA (5 V DC)	max. 1600 mA (5 V DC)	max. 1600 mA (5 V DC)
Verfügbarer Strom (24 V DC)	max. 400 mA (Geberversorgung)	max. 400 mA (Geberversorgung)	max. 400 mA (Geberversorgung)
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang	4 mA/Eingang	4 mA/Eingang

Tabelle A- 45 CPU-Merkmale

Technische Daten	Beschreibung	
Anwenderspeicher <sup>1</sup>	Arbeitsspeicher	75 KB
	Last	4 MB intern, erweiterbar bis auf SD-Kartengröße
	Remanent	10 KB
Integrierte digitale E/A	14 Eingänge/10 Ausgänge	
Integrierte analoge E/A	2 Eingänge	
Größe des Prozessabbilds	1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)	
Merker (M)	8192 Byte	
Temporärer (lokaler) Speicher	<ul style="list-style-type: none"> <li>• 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs)</li> <li>• 4 KB für Standardalarmereignisse, einschließlich FBs und FCs</li> <li>• 4 KB für Fehleralarmereignisse, einschließlich FBs und FCs</li> </ul>	
Zusätzliche Signalmodule	max. 8 SMs	

Technische Daten	Beschreibung
Erweiterung SB, CB, BB	max. 1
Zusätzliche Kommunikationsmodule	max. 3 CM
Schnelle Zähler	6 insgesamt, siehe Tabelle Funktionsweise von schnellen Zählern (Seite 359) <ul style="list-style-type: none"> <li>• Einphasenzähler: 3 bei 100 kHz und 3 bei 30 kHz Takt,</li> <li>• A/B-Zähler: 3 bei 80 kHz und 3 bei 20 kHz Takt</li> </ul>
Impulsausgänge <sup>2</sup>	4
Eingänge für Impulsabgriff	14
Verzögerungs-/Weckalarme	4 mit Auflösung von 1 ms
Flankenalarme	12 steigend und 12 fallend (14 und 14 mit optionalem Signalboard)
Memory Card	SIMATIC Memory Card (optional)
Genauigkeit Echtzeituhr	+/- 60 Sekunden/Monat
Pufferung Echtzeituhr	Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

<sup>1</sup> Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU begrenzt. Die Anzahl der unterstützten OBs, FCs, FBs und DBs und die Größe der einzelnen Bausteine ist nicht begrenzt. Die einzige Begrenzung ist die Gesamtspeicherkapazität.

<sup>2</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Tabelle A- 46 Leistung

Art der Anweisung	Ausführungsgeschwindigkeit
Boolesch	0,08 µs/Operation
Wort übertragen	1,7 µs/Operation
Realzahlenarithmetik	2,3 µs/Operation

## A.4.2 Von der CPU 1214C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A- 47 Von der CPU 1214C unterstützte Bausteine, Zeiten und Zähler

Element	Beschreibung	
Bausteine	Typ	OB, FB, FC, DB
	Größe	64 KB
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	1 bis 65535 (z. B. FB 1 bis FB 65535)
	Schachteltiefe	16 aus Zyklus- oder Anlauf-OBs, 4 aus Verzögerungsalarm-, Uhrzeitalarm-, Weckalarm-, Prozessalarm-, Zeitfehler- oder Diagnosefehler-OBs
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.
	OBs	Programmzyklus
	Mehrere: OB 1, OB 200 bis OB 65535	



Tabelle A- 49 Spannungsversorgung

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Spannungsbereich	85 bis 264 V AC	20,4 V DC bis 28,8 V DC 22,0 VDC bis 28,8 V DC bei Umgebungstemperatur -20 °C bis 0 °C	
Netzfrequenz	47 bis 63 Hz	--	
Einschaltstrom (max. Last)	Nur CPU	100 mA bei 120 V AC 50 mA bei 240 V AC	500 mA bei 24 V DC
	CPU mit allen Erweiterungsbaugruppen	300 mA bei 120 V AC 150 mA bei 240 V AC	1500 mA bei 24 V DC
Einschaltstrom (max.)	20 A bei 264 V AC	12 A bei 28,8 V DC	
Elektrische Trennung (Eingangsleistung zu Logik)	1500 V AC	Nicht elektrisch getrennt	
Kriechstrom an Erde, AC-Leitung an Funktionserde	max. 0.5 mA	-	
Verzögerungszeit (Spannungsverlust)	20 ms bei 120 V AC 80 ms bei 240 V AC	10 ms bei 24 V DC	
Interne Sicherung, nicht durch Anwender austauschbar	3 A, 250 V, träge		

Tabelle A- 50 Geberversorgung

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Spannungsbereich	20,4 bis 28,8 V DC	L + minus 4 V DC min. L+ minus 5 V DC min. bei einer Umgebungstemperatur von -20 °C bis 0 °C	
Nennausgangsstrom (max.)	400 mA (kurzschlussfest)		
Max. Welligkeit/Störströme (<10 MHz)	< 1 V SpitzeSpitze	Wie Eingangsleitung	
Elektrische Trennung (CPU-Logik zu Sensorversorgung)	Nicht elektrisch getrennt		

### A.4.3 Digitale Eingänge und Ausgänge

Tabelle A- 51 Digitaleingänge

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Anzahl der Eingänge	14		
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)		
Nennspannung	24 V DC bei 4 mA, Nennwert		
Zulässige Dauerspannung	max. 30 V DC		
Stoßspannung	35 V DC für 0,5 s		
Signal logisch 1 (min.)	15 V DC bei 2,5 mA		
Signal logisch 0 (max.)	5 V DC bei 1 mA		
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute		
Potenzialgetrennte Gruppen	1		
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)		
HSC Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	Einphasenzähler: 100 kHz (Ea.0 bis Ea.5) und 30 kHz (Ea.6 bis Eb.5) A/B-Zähler: 80 kHz (Ea.0 bis Ea.5) und 20 kHz (Ea.6 bis Eb.5)		
Anzahl gleichzeitig eingeschalteter Eingänge	<ul style="list-style-type: none"> <li>• 7 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 14 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>		
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge		

Tabelle A- 52 Digitale Ausgänge

Technische Daten	CPU 1214C AC/DC/Relais und DC/DC/Relais	CPU 1214C DC/DC/DC
Ausgänge	10	10
Typ	Relais, Trockenkontakt	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule zu Kontakt) Keine (Spule zu Logik)	500 V AC für 1 Minute
Isolationswiderstand	min. 100 MΩ, wenn neu	--
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	--

Technische Daten	CPU 1214C AC/DC/Relais und DC/DC/Relais	CPU 1214C DC/DC/DC
Potentialgetrennte Gruppen	2	1
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 µs von Aus nach Ein max. 3,0 µs von Ein nach Aus
Schaltverzögerung (Aa.4 bis Ab.1)	max. 10 ms	max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	--
Frequenz Impulsgenerator (Aa.0 und Aa.2)	Nicht empfehlenswert <sup>1</sup>	max. 100 kHz, min. 2 Hz <sup>2</sup>
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> <li>• 5 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 10 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	

<sup>1</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

<sup>2</sup> Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

## A.4.4 Analoge Eingänge

Tabelle A- 53 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27.649 bis 32.511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32.512 bis 32.767
Auflösung	10 Bits
Max. Stehspannung	35 V DC

Technische Daten	Beschreibung
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 776).
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25°C / -20 bis 60°C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrilltes Leiterpaar

#### A.4.4.1 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A- 54 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
<b>Abtastzeit</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

#### A.4.4.2 Abtastzeit der integrierten analogen Ports der CPU

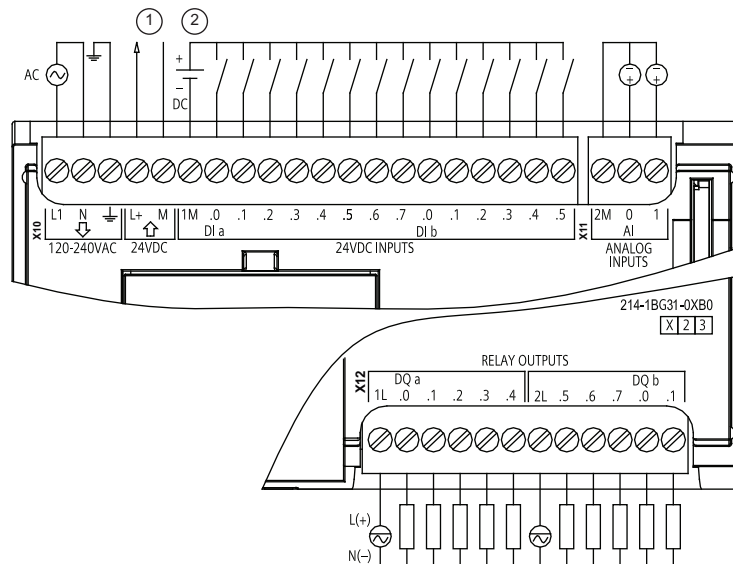
Tabelle A- 55 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms



### A.4.5 Schaltpläne der CPU 1214C

Tabelle A- 56 CPU 1214C AC/DC/Relais (6ES7 214-1BG31-0XB0)



- ① 24-V-DC-  
Geberspannung  
Um zusätzliche  
Störfestigkeit zu  
erreichen, schließen  
Sie "M" an Masse an,  
auch wenn Sie keine  
Geberspannung  
verwenden.
- ② Bei stromziehenden  
Eingängen "-" an "M"  
anschließen (gezeigt).  
Bei stromliefernden  
Eingängen "+" an "M"  
anschließen.

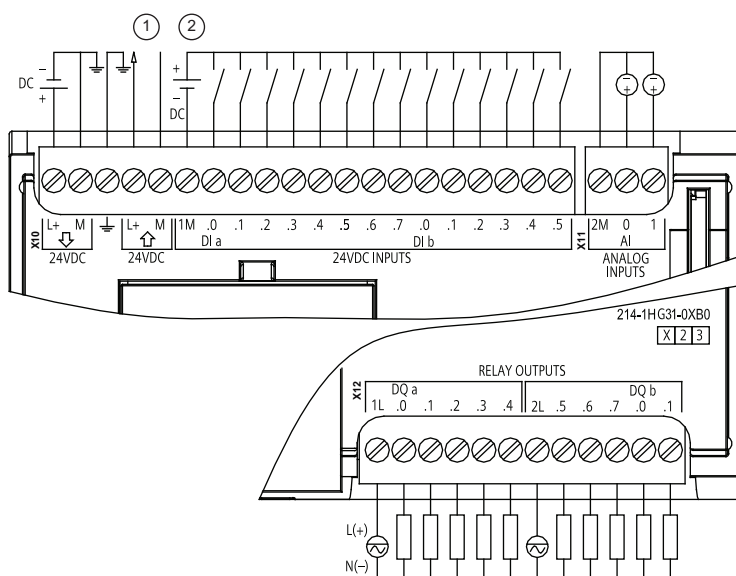
Hinweis: X11-  
Steckverbinder müssen  
vergoldet sein. Im Anhang  
C, Ersatzteile, finden Sie  
die Bestellnummer.

Tabelle A- 57 Anschlussbelegung für die CPU 1214C AC/DC/Relais (6ES7 214-1BG31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / 24-V-DC- Sensorausgang	--	DO a.2
5	M / 24-V-DC- Sensorausgang	--	DO a.3
6	1M	--	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--

Pin	X10	X11 (vergoldet)	X12
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tabelle A- 58 CPU 1214C DC/DC/Relais (6ES7 214-1HG31-0XB0)



- ① 24-V-DC-Geberspannung  
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberspannung verwenden.
- ② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).  
Bei stromliefernden Eingängen "+" an "M" anschließen.

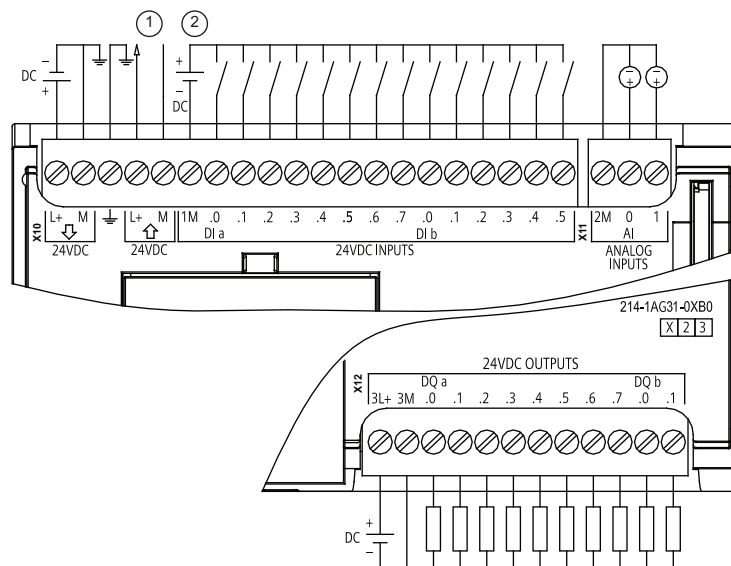
Hinweis: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 59 Anschlussbelegung für die CPU 1214C DC/DC/Relais (6ES7 214-1HG31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / 24-V-DC-Sensorausgang	--	DO a.2
5	M / 24-V-DC-Sensorausgang	--	DO a.3
6	1M	--	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7

Pin	X10	X11 (vergoldet)	X12
11	DI a.4	--	DO b.0
12	DI a.5	--	DO b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tabelle A- 60 CPU 1214C DC/DC/DC (6ES7 214-1AG31-0XB0)



- ① 24-V-DC-  
Geberspannung  
Um zusätzliche  
Störfestigkeit zu  
erreichen, schließen  
Sie "M" an Masse an,  
auch wenn Sie keine  
Gebersversorgung  
verwenden.
- ② Bei stromziehenden  
Eingängen "-" an "M"  
anschießen  
(abgebildet).  
Bei stromliefernden  
Eingängen "+" an "M"  
anschießen.

Hinweis: X11-  
Steckverbinder müssen  
vergoldet sein. Im Anhang  
C, Ersatzteile, finden Sie  
die Bestellnummer.

Tabelle A- 61 Anschlussbelegung für die CPU 1214C DC/DC/DC (6ES7 214-1AG31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Funktionserde	AI 1	DO a.0
4	L+ / 24-V-DC- Sensorausgang	--	DO a.1

Pin	X10	X11 (vergoldet)	X12
5	M / 24-V-DC-Sensorausgang	--	DO a.2
6	1M	--	DO a.3
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO b.1
13	DI a.6	--	--
14	DI a.7	--	-
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

**Hinweis**

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

## A.5 CPU 1215C

### A.5.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A- 62 Allgemein

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Bestellnummer	6ES7 215-1BG31-0XB0	6ES7 215-1HG31-0XB0	6ES7 215-1AG31-0XB0
Abmessungen (B x H x T) (mm)	130 x 100 x 75	130 x 100 x 75	130 x 100 x 75
Versandgewicht	550 Gramm	585 Gramm	520 Gramm
Leistungsverlust	14 W	12 W	12 W
Verfügbarer Strom (SM- und CM-Bus)	max. 1600 mA (5 V DC)	max. 1600 mA (5 V DC)	max. 1600 mA (5 V DC)

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Verfügbarer Strom (24 V DC)	max. 400 mA (Geberversorgung)	max. 400 mA (Geberversorgung)	max. 400 mA (Geberversorgung)
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang	4 mA/Eingang	4 mA/Eingang

Tabelle A- 63 CPU-Merkmale

Technische Daten	Beschreibung	
Anwenderspeicher <sup>1</sup>	Arbeitsspeicher	100 KB
	Ladespeicher	4 MB, intern, erweiterbar bis auf SD-Kartengröße
	Remanent	10 KB
Integrierte digitale E/A	14 Eingänge/10 Ausgänge	
Integrierte analoge E/A	2 Eingänge/2 Ausgänge	
Größe des Prozessabbilds	1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)	
Merker (M)	8192 Byte	
Temporärer (lokaler) Speicher	<ul style="list-style-type: none"> <li>• 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs)</li> <li>• 4 KB für Standardalarmereignisse, einschließlich FBs und FCs</li> <li>• 4 KB für Fehleralarmereignisse, einschließlich FBs und FCs</li> </ul>	
Zusätzliche Signalmodule	max. 8 SMs	
Erweiterung SB, CB, BB	max. 1	
Zusätzliche Kommunikationsmodule	max. 3 CM	
Schnelle Zähler	6 insgesamt, siehe Tabelle HSC-Eingangszuweisungen bei der CPU 1215C <ul style="list-style-type: none"> <li>• Einphasenzähler: 3 bei 100 kHz und 3 bei 30 kHz Takt</li> <li>• A/B-Zähler: 3 bei 80 kHz und 3 bei 20 kHz Takt</li> </ul>	
Impulsausgänge <sup>2</sup>	4	
Eingänge für Impulsabgriff	14	
Verzögerungs-/Weckalarme	4 mit Auflösung von 1 ms	
Flankenalarme	12 steigend und 12 fallend (14 und 14 mit optionalem Signalboard)	
Memory Card	SIMATIC Memory Card (optional)	
Genauigkeit Echtzeituhr	+/- 60 Sekunden/Monat	
Pufferung Echtzeituhr	Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)	

<sup>1</sup> Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU begrenzt. Die Anzahl der unterstützten OBs, FCs, FBs und DBs und die Größe der einzelnen Bausteine ist nicht begrenzt. Die einzige Begrenzung ist die Gesamtspeicherkapazität.

<sup>2</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.



Tabelle A- 66 Kommunikation

Technische Daten	Beschreibung
Schnittstellen	2
Typ	Ethernet
HMI-Gerät <sup>1</sup>	3
Programmiergerät (PG)	1
Anschlüsse	<ul style="list-style-type: none"> <li>• 8 für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV</li> <li>• 3 für die S7-Kommunikation über Server-GET/PUT (CPU-zu-CPU)</li> <li>• 8 für die S7-Kommunikation über Client-GET/PUT (CPU-zu-CPU)</li> </ul>
Datenraten	10/100 MBit/s
Elektrische Trennung (externes Signal zu PLC-Logik)	Wandler potentialgetrennt, 1500 V AC, nur für kurzfristige Sicherheit
Kabelart	CAT5e geschirmt

<sup>1</sup> Die CPU bietet zweckbestimmte HMI-Verbindungen, um bis zu 3 HMI-Geräte zu unterstützen. (Sie können bis zu 2 SIMATIC Comfort Panels haben.) Wie viele HMI-Geräte insgesamt unterstützt werden, hängt von den Typen der HMI-Panels in Ihrer Konfiguration ab. Sie können beispielsweise bis zu drei SIMATIC Basic Panels an Ihre CPU anschließen, oder Sie können bis zu zwei SIMATIC Comfort Panels und ein zusätzliches Basic Panel anschließen.

Tabelle A- 67 Spannungsversorgung

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Spannungsbereich	85 bis 264 V AC	20,4 V DC bis 28,8 V DC 22,0 VDC bis 28,8 V DC bei Umgebungstemperatur -20 °C bis 0 °C	
Netzfrequenz	47 ... 63 Hz	--	
Einschaltstrom (max. Last)	Nur CPU 100 mA bei 120 V AC 50 mA bei 240 V AC	500 mA bei 24 V DC	
	CPU mit allen Erweiterungsbaugruppen 300 mA bei 120 V AC 150 mA bei 240 V AC	1500 mA bei 24 V DC	
Einschaltstrom (max.)	20 A bei 264 V AC	12 A bei 28,8 V DC	
Elektrische Trennung (Eingangsleistung zu Logik)	1500 V AC	Nicht elektrisch getrennt	
Kriechstrom an Erde, AC-Leitung an Funktionserde	max. 0.5 mA	-	
Verzögerungszeit (Spannungsverlust)	20 ms bei 120 V AC 80 ms bei 240 V AC	10 ms bei 24 V DC	
Interne Sicherung, nicht durch Anwender austauschbar	3 A, 250 V, träge		

A.5 CPU 1215C

Tabelle A- 68 Geberversorgung

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Spannungsbereich	20,4 bis 28,8 V DC	L + minus 4 V DC min. L+ minus 5 V DC min. bei einer Umgebungstemperatur von -20 °C bis 0 °C	
Nennausgangsstrom (max.)	400 mA (kurzschlussfest)		
Max. Welligkeit/Störströme (<10 MHz)	< 1 V Spitze-Spitze	Wie Eingangsleitung	
Elektrische Trennung (CPU-Logik zu Sensorversorgung)	Nicht elektrisch getrennt		

### A.5.3 Digitale Eingänge und Ausgänge

Tabelle A- 69 Digitaleingänge

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Anzahl der Eingänge	14		
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)		
Nennspannung	24 V DC bei 4 mA, Nennwert		
Zulässige Dauerspannung	max. 30 V DC		
Stoßspannung	35 V DC für 0,5 s		
Signal logisch 1 (min.)	15 V DC bei 2,5 mA		
Signal logisch 0 (max.)	5 V DC bei 1 mA		
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute		
Potentialgetrennte Gruppen	1		
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)		
HSC Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	Einphasenzähler: 100 kHz (Ea.0 bis Ea.5) und 30 kHz (Ea.6 bis Eb.5) A/B-Zähler: 80 kHz (Ea.0 bis Ea.5) und 20 kHz (Ea.6 bis Eb.5)		
Anzahl gleichzeitig eingeschalteter Eingänge	<ul style="list-style-type: none"> <li>• 7 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 14 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>		
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge		

Tabelle A- 70 Digitale Ausgänge

Technische Daten	CPU 1215C AC/DC/Relais und CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Ausgänge	10	10
Typ	Relais, Trockenkontakt	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC



Technische Daten	CPU 1215C AC/DC/Relais und CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 k $\Omega$ Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 $\Omega$ wenn neu	max. 0,6 $\Omega$
Kriechstrom pro Ausgang	--	max. 10 $\mu$ A
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule zu Kontakt) Keine (Spule zu Logik)	500 V AC für 1 Minute
Isolationswiderstand	min. 100 M $\Omega$ , wenn neu	--
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	--
Potentialgetrennte Gruppen	2	1
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 $\mu$ s von Aus nach Ein max. 3,0 $\mu$ s von Ein nach Aus
Schaltverzögerung (Aa.4 bis Ab.1)	max. 10 ms	max. 50 $\mu$ s von Aus nach Ein max. 200 $\mu$ s von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	--
Frequenz Impulsgenerator (Aa.0 und Aa.2)	Nicht empfehlenswert <sup>1</sup>	max. 100 kHz, min. 2 Hz <sup>2</sup>
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> <li>• 5 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 10 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	

<sup>1</sup> Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

<sup>2</sup> Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

## A.5.4 Analogeingänge und -ausgänge

### A.5.4.1 Technische Daten der analogen Eingänge

Tabelle A- 71 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27.649 bis 32.511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32.512 bis 32.767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU.
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25°C / -20 bis 60°C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrehtes Leiterpaar

### A.5.4.2 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A- 72 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
<b>Abtastzeit</b>	<b>4,17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

### A.5.4.3 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A- 73 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

### A.5.4.4 Technische Daten der Analogausgänge

#### Analoge Ausgänge

Tabelle A- 74 Analoge Ausgänge

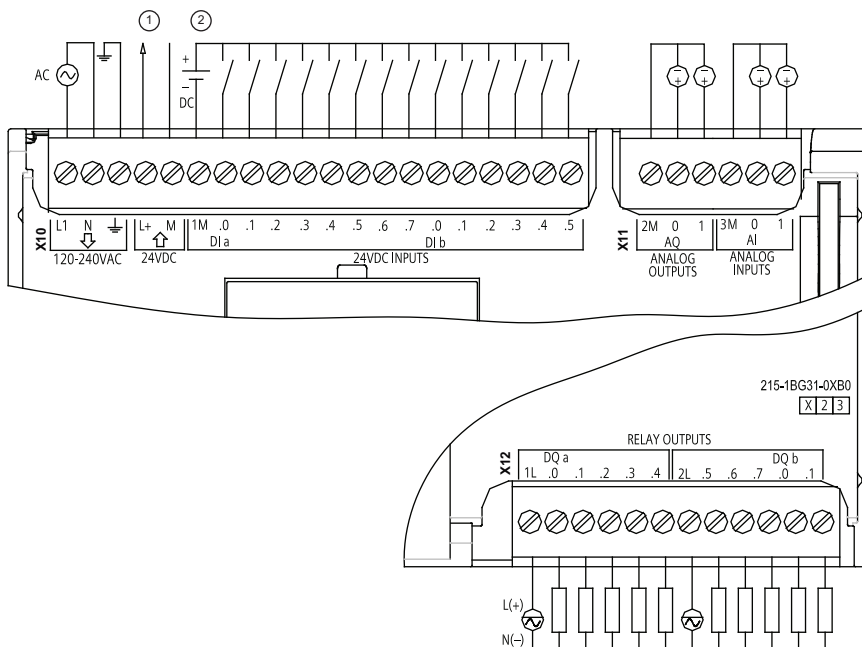
Technische Daten	Beschreibung
Ausgänge	2
Typ	Strom
Vollausschlagsbereich	0 ... 20 mA
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	20,01 bis 23,52 mA <sup>1</sup>
Überschwingbereich (Datenwort)	27.649 bis 32.511
Überlaufbereich	siehe Fußnote <sup>2</sup>
Datenwort Überlaufbereich	32.512 bis 32.767
Auflösung	10 Bit
Ausgangsimpedanz	≤ 500 Ω max.
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60°C)	3,0 % / 3,5 % des Vollausschlags
Einschwingzeit	2 ms
Leitungslänge (Meter)	100 m, geschirmtes, verdrehtes Leiterpaar

<sup>1</sup> Bei der CPU 1215C mit DC-Versorgung: Bei einer Versorgungsspannung von 20,4 V DC wird im Überschwingbereich eine Ausgangsimpedanz bis 400 Ω unterstützt.

<sup>2</sup> Bei Überlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: Ersatzwert aufschalten oder Letzten Wert halten.

### A.5.5 Schaltpläne der CPU 1215C

Tabelle A- 75 CPU 1215C AC/DC/Relais (6ES7 215-1BG31-0XB0)



- ① 24-V-DC-Geberspannung  
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Gebersversorgung verwenden.
- ② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).  
Bei stromliefernden Eingängen "+" an "M" anschließen.

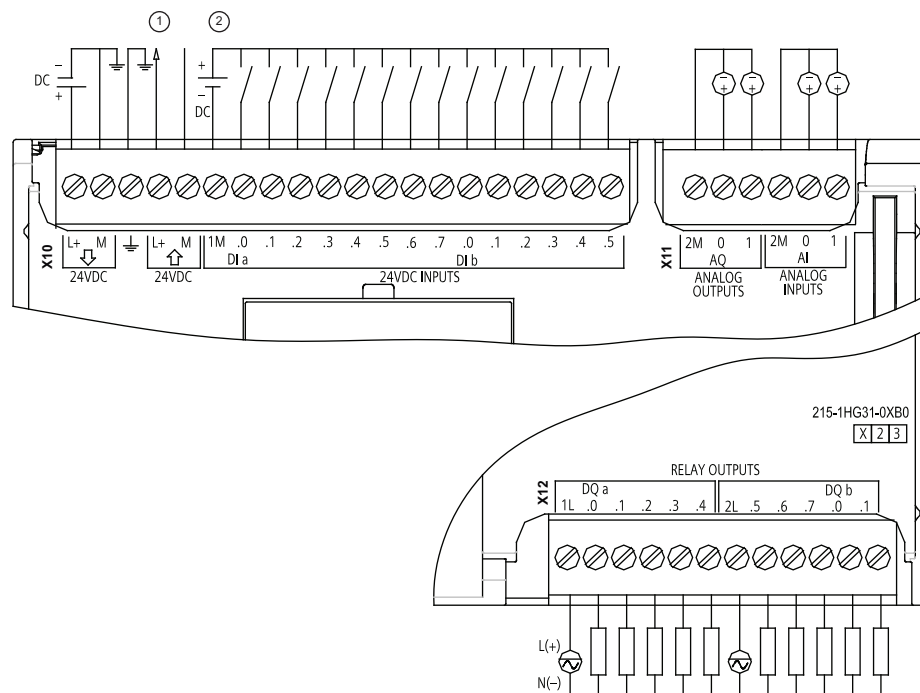
Hinweis: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 76 Anschlussbelegung für die CPU 1215C AC/DC/Relais (6ES7 215-1BG31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AO 0	DO a.0
3	Funktionserde	AO 1	DO a.1
4	L+ / 24-V-DC-Sensorausgang	3M	DO a.2
5	M / 24-V-DC-Sensorausgang	AI 0	DO a.3
6	1M	AI 1	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0

Pin	X10	X11 (vergoldet)	X12
12	DI a.5	--	DO b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tabelle A- 77 CPU 1215C DC/DC/Relais (6ES7 215-1HG31-0XB0)



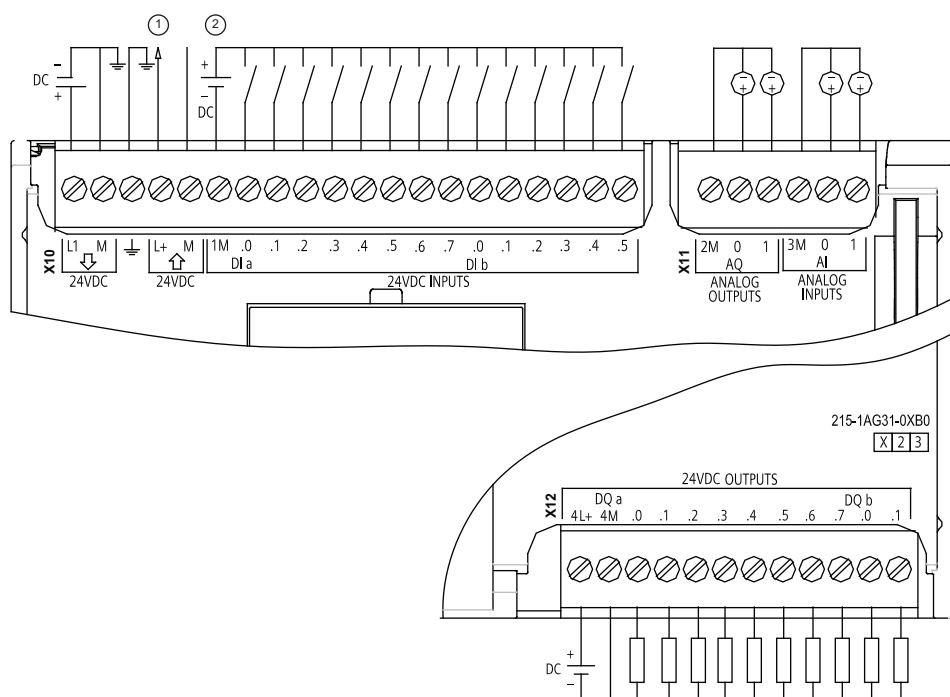
- ① 24-V-DC-Geberverspannung  
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.
- ② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet). Bei stromliefernden Eingängen "+" an "M" anschließen.

Hinweis: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 78 Anschlussbelegung für die CPU 1215C DC/DC/Relais (6ES7 215-1HG31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AO 0	DO a.0
3	Funktionserde	AO 1	DO a.1
4	L+ / 24-V-DC-Sensorausgang	2M	DO a.2
5	M / 24-V-DC-Sensorausgang	AI 0	DO a.3
6	1M	AI 1	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Tabelle A- 79 CPU 1215C DC/DC/DC (6ES7 215-1AG31-0XB0)



- ① 24-V-DC-Geberverspannung  
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.
- ② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet). Bei stromliefernden Eingängen "+" an "M" anschließen.

Hinweis: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 80 Anschlussbelegung für die CPU 1215C DC/DC/DC (6ES7 215-1AG31-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 24 V DC	2 M	4L+
2	M / 24 V DC	AO 0	4M
3	Funktionserde	AO 1	DO a.0
4	L+ / 24-V-DC-Sensorausgang	3M	DO a.1
5	M / 24-V-DC-Sensorausgang	AI 0	DO a.2
6	1M	AI 1	DO a.3
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO b.1

Pin	X10	X11 (vergoldet)	X12
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

**Hinweis**

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

## A.6 Digitale Signalmodule (SM)

### A.6.1 Technische Daten für das digitale Eingangsmodul SM 1221

Tabelle A- 81 Allgemeine technische Daten

Modell	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Bestellnummer	6ES7 221-1BF30-0XB0	6ES7 221-1BH30-0XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	170 Gramm	210 Gramm
Leistungsverlust	1,5 W	2,5 W
Stromaufnahme (SM-Bus)	105 mA	130 mA
Stromaufnahme (24 V DC)	4 mA / Eingang	4 mA / Eingang

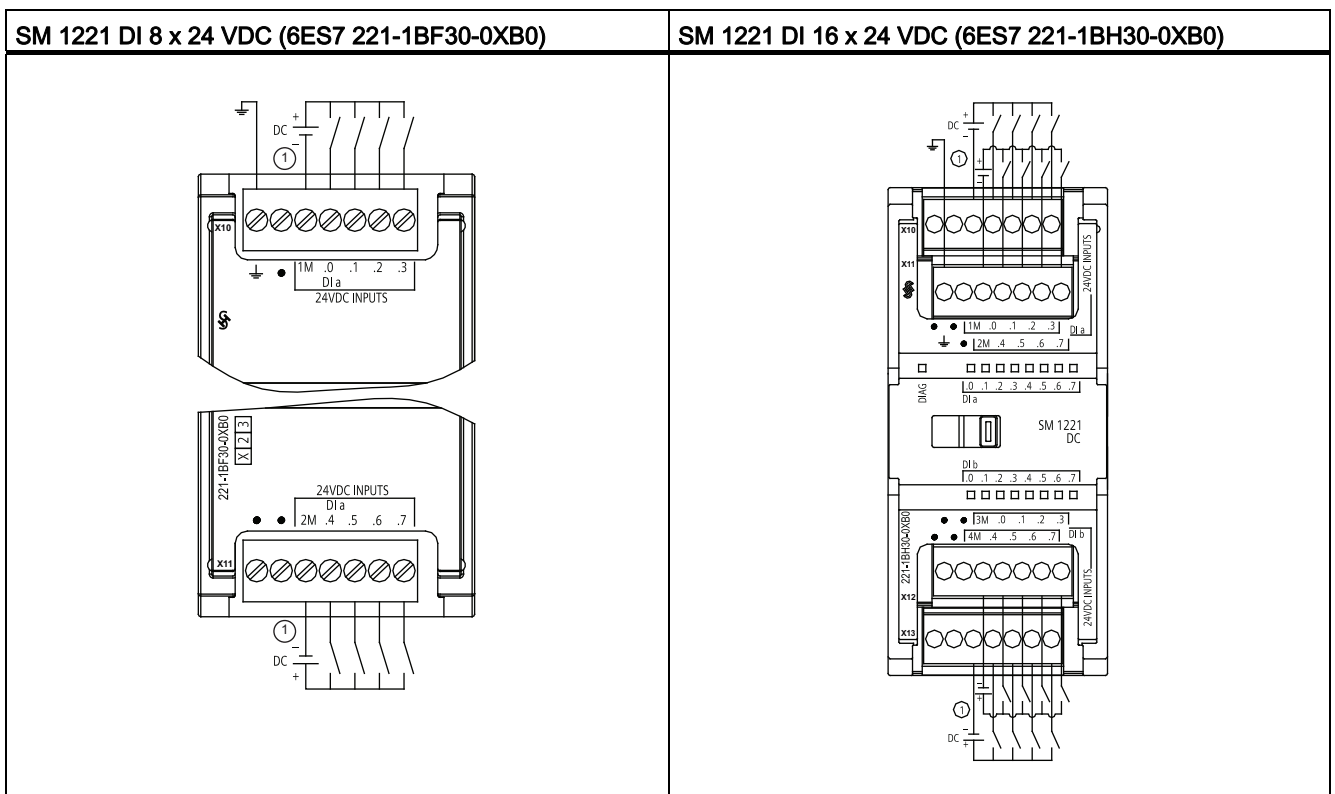
Tabelle A- 82 Digitaleingänge

Modell	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Anzahl der Eingänge	8	16
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)
Nennspannung	24 V DC bei 4 mA, Nennwert	24 V DC bei 4 mA, Nennwert
Zulässige Dauerspannung	max. 30 V DC	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA	15 V DC bei 2,5 mA



Modell	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Signal logisch 0 (max.)	5 V DC bei 1 mA	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute	500 V AC für 1 Minute
Potentialgetrennte Gruppen	2	4
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)
Anzahl gleichzeitig eingeschalteter Eingänge	8	16
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt	500 m geschirmt, 300 m ungeschirmt

Tabelle A- 83 Schaltpläne der Digitaleingangs-SMs



① Bei stromziehenden Eingängen "-" an "M" anschließen (gezeigt). Bei stromliefernden Eingängen "+" an "M" anschließen.

Tabelle A- 84 Anschlussbelegung für das SM 1221 DI 8 x 24 VDC (6ES7 221-1BF30-0XB0)

Pin	X10	X11
1	GND	Kein Anschluss
2	Kein Anschluss	Kein Anschluss
3	1M	2M
4	DO a.0	DI a.4
5	DI a.1	DI a.5
6	DI a.2	DI a.6
7	DI a.3	DI a.7

Tabelle A- 85 Anschlussbelegung für das SM 1221 DI 16 x 24 VDC (6ES7 221-1BH30-0XB0)

Pin	X10	X11	X12	X13
1	GND	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1M	2M	3 M	4 M
4	DO a.0	DI a.4	DI b.0	DI b.4
5	DI a.1	DI a.5	DI b.1	DI b.5
6	DI a.2	DI a.6	DI b.2	DI b.6
7	DI a.3	DI a.7	DI b.3	DI b.7

## A.6.2 Technische Daten für das digitale Ausgangsmodul SM 1222 mit 8 Ausgängen

Tabelle A- 86 Allgemeine technische Daten

Modell	SM 1222 DO 8 x Relais	SM 1222 DO8 RLS, Umschaltung	SM 1222 DO 8 x 24 V DC
Bestellnummer	6ES7 222-1HF30-0XB0	6ES7 222-1XF30-0XB0	6ES7 222-1BF30-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
Gewicht	190 Gramm	310 Gramm	180 Gramm
Leistungsverlust	4,5 W	5 W	1,5 W
Stromaufnahme (SM-Bus)	120 mA	140 mA	120 mA
Stromaufnahme (24 V DC)	11 mA / Relaispule	16,7 mA / Relaispule	11 mA / Relaispule

Tabelle A- 87 Digitale Ausgänge

Modell	SM 1222 DO 8 x Relais	SM 1222 DO8 RLS, Umschaltung	SM 1222 DO 8 x 24 V DC
Ausgänge	8	8	8
Typ	Relais, Trockenkontakt	Relaiswechselkontakt	MOSFET, elektronisch (stromliefernd)

Modell	SM 1222 DO 8 x Relais	SM 1222 DO8 RLS, Umschaltung	SM 1222 DO 8 x 24 V DC
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	--	min. 20 V DC
Signal logisch 0 bei 10 k $\Omega$ Last	--	--	max. 0,1 V DC
Strom (max.)	2,0 A	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	30 W DC/200 W AC	5 W
Kontaktwiderstand bei EIN	max. 0,2 $\Omega$ wenn neu	max. 0,2 $\Omega$ wenn neu	max. 0,6 $\Omega$
Kriechstrom pro Ausgang	--	--	max. 10 $\mu$ A
Einschaltstrom	7 A bei geschlossenen Kontakten	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule-Kontakt) Keine (Spule-Logik)	1500 V AC für 1 Minute (Spule zu Kontakt)	500 V AC für 1 Minute
Isolationswiderstand	min. 100 M $\Omega$ , wenn neu	min. 100 M $\Omega$ , wenn neu	--
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	750 V AC für 1 Minute	--
Potenzialgetrennte Gruppen	2	8	1
Strom je Leiter (max.)	10 A	2 A	4 A
Induktive Klemmspannung	--	--	L+ minus 48 V, 1 W Verlustleistung
Schaltverzögerung	max. 10 ms	max. 10 ms	max. 50 $\mu$ s von Aus nach Ein max. 200 $\mu$ s von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	1 Hz	--
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Anzahl gleichzeitig eingeschalteter Ausgänge	8	<ul style="list-style-type: none"> <li>• 4 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 8 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	8
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt

### A.6.3 Technische Daten für das digitale Ausgangsmodul SM 1222 mit 16 Ausgängen

Tabelle A- 88 Allgemeine technische Daten

Modell	SM 1222 DO 16 x Relais	SM 1222 DO 16 x 24 V DC
Bestellnummer	6ES7 222-1HH30-0XB0	6ES7 222-1BH30-0XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	260 Gramm	220 Gramm
Leistungsverlust	8,5 W	2,5 W
Stromaufnahme (SM-Bus)	135 mA	140 mA
Stromaufnahme (24 V DC)	11 mA / Relaisspule	-

Tabelle A- 89 Digitale Ausgänge

Modell	SM1222 DO 16 x Relais	SM1222 DO 16 x 24 V DC
Ausgänge	16	16
Typ	Relais, Trockenkontakt	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	-	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	-	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 μA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule-Kontakt) Keine (Spule-Logik)	500 V AC für 1 Minute
Isolationswiderstand	min. 100 MΩ, wenn neu	-
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	-
Potenzialgetrennte Gruppen	4	1
Strom je Leiter (max.)	10 A	8 A
Induktive Klemmspannung	-	L+ minus 48 V, 1 W Verlustleistung
Schaltverzögerung	max. 10 ms	max. 50 μs von Aus nach Ein max. 200 μs von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	-
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	-

Modell	SM1222 DO 16 x Relais	SM1222 DO 16 x 24 V DC
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	-
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> <li>• 8 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 16 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	16
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt

Tabelle A- 90 Schaltpläne der Digitalausgangs-SMs mit 8 Ausgängen

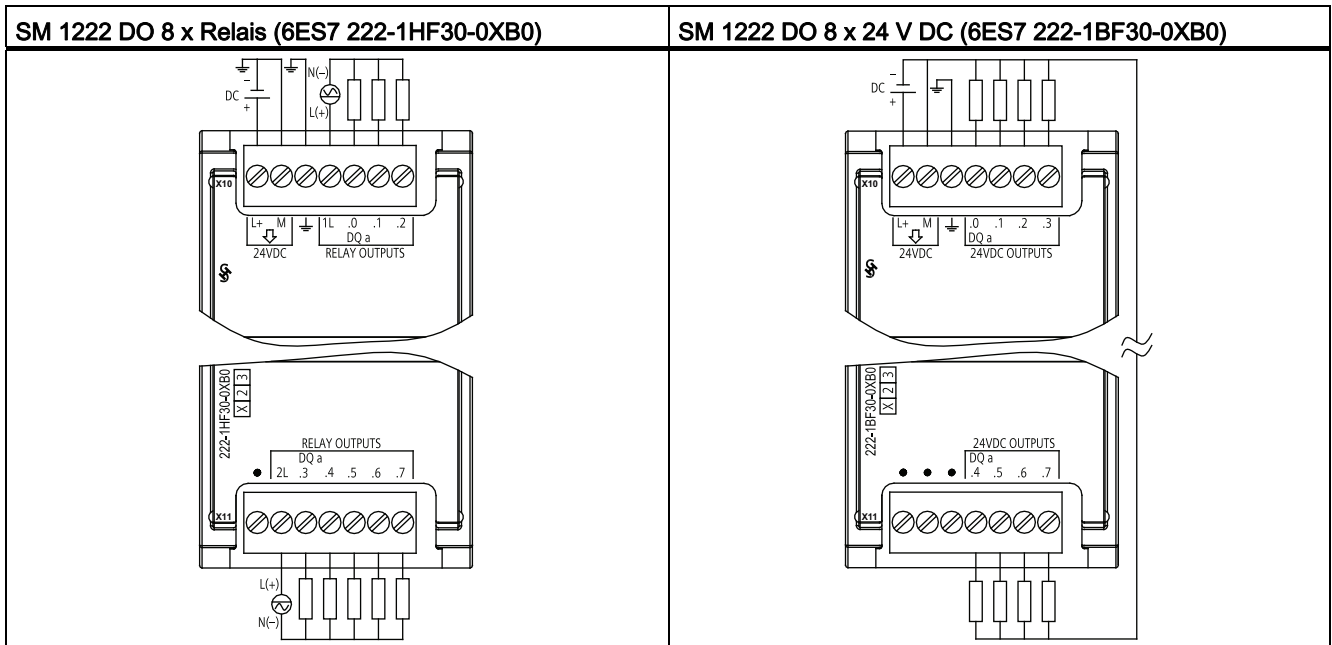


Tabelle A- 91 Anschlussbelegung für das SM 1222 DO 8 x Relais (6ES7 222-1HF30-0XB0)

Pin	X10	X11
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	2L
3	Funktionserde	DO a.3
4	1L	DO a.4
5	DO a.0	DO a.5
6	DO a.1	DO a.6
7	DO a.2	DO a.7

Tabelle A- 92 Anschlussbelegung für das SM 1222 DO 8 x 24 VDC (6ES7 222-1BF30-0XB0)

Pin	X10	X11
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	Funktionserde	Kein Anschluss
4	DO a.0	DO a.4
5	DO a.1	DO a.5
6	DO a.2	DO a.6
7	DO a.2	DO a.7

Tabelle A- 93 Schaltplan für das Umschaltungs-Digitalausgangs-SM mit 8 Relaisausgängen

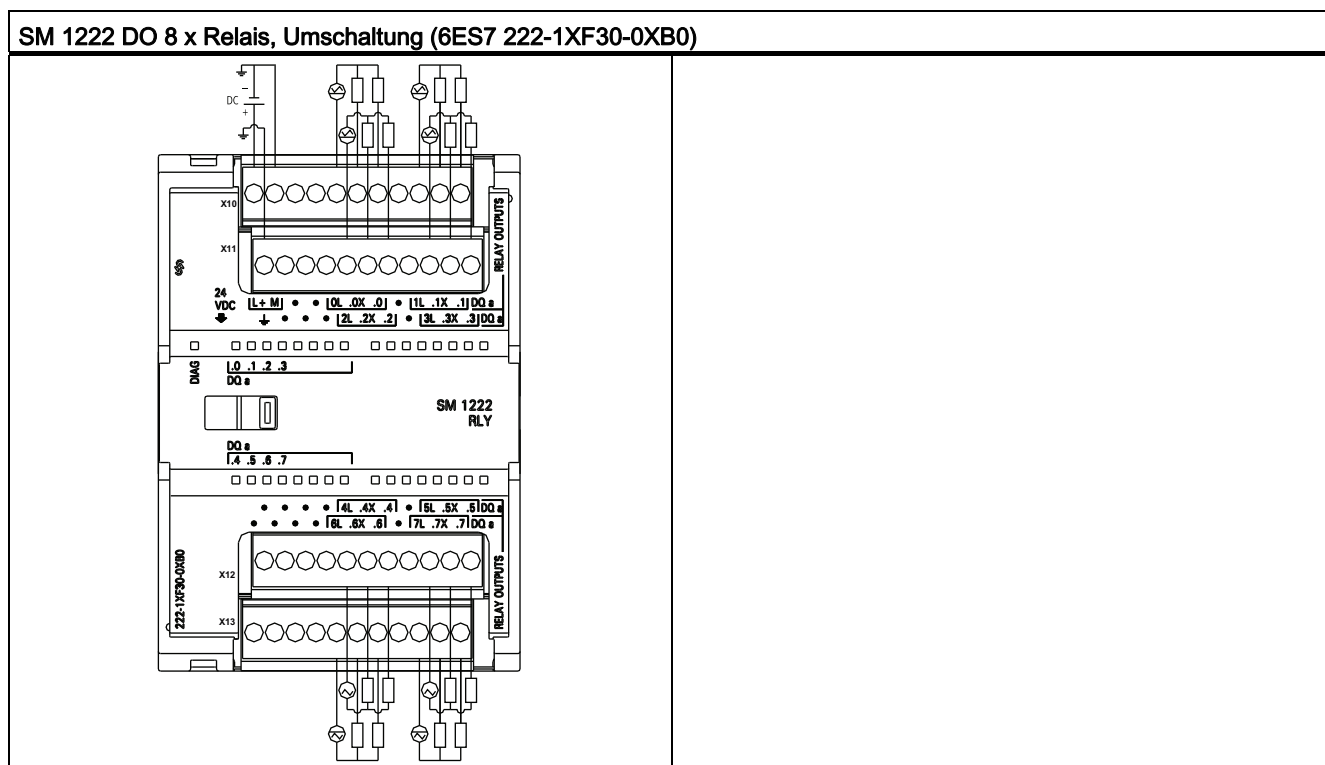


Tabelle A- 94 Anschlussbelegung für das SM 1222 DO 8 x Relais, Umschaltung (6ES7 222-1XF30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss

Pin	X10	X11	X12	X13
4	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
5	0L	2L	4L	6L
6	DO a.0X	DO a.2X	DO a.4X	DO a.6X
7	DO a.0	DO a.2	DO a.4	DO a.6
8	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
9	1L	3L	5L	7L
10	DO a.1X	DO a.3X	DO a.5X	DO a.7X
11	DO a.1	DO a.3	DO a.5	DO a.7

Tabelle A- 95 Schaltpläne der Digitalausgangs-SMs mit 16 Ausgängen

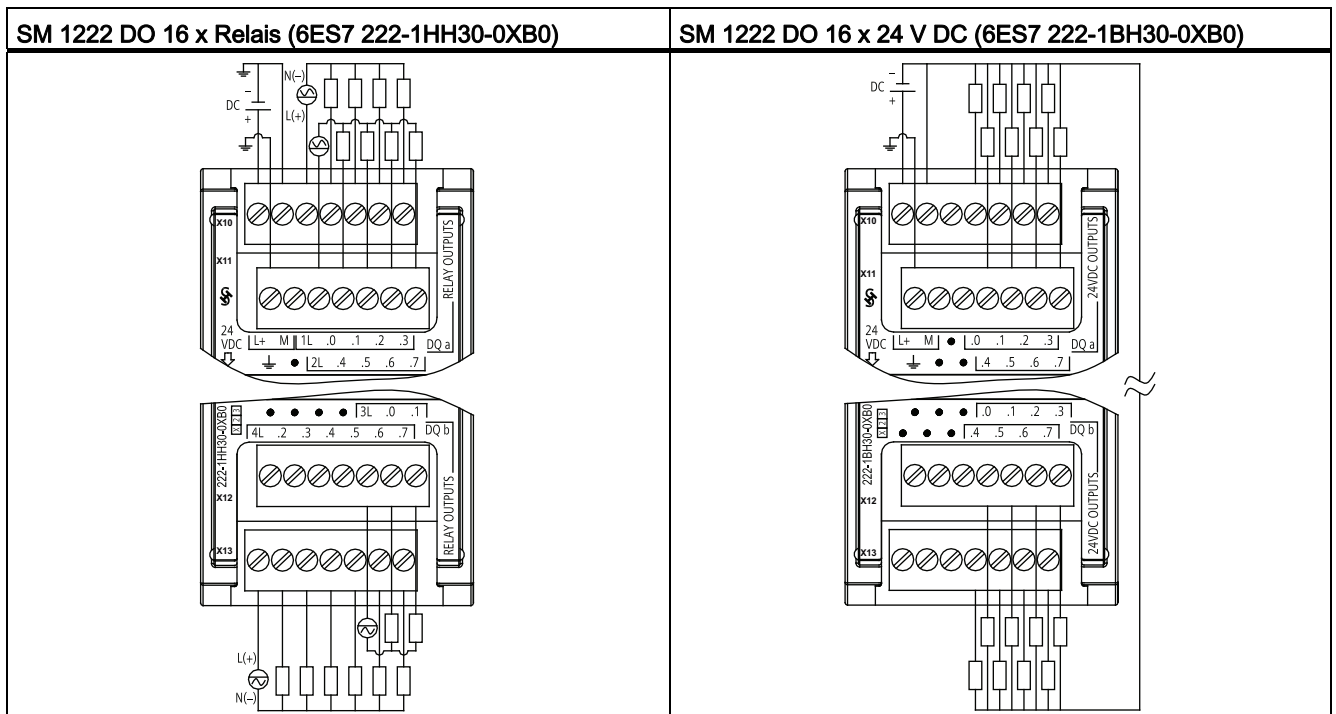


Tabelle A- 96 Anschlussbelegung für das SM 1222 DO 16 x Relais (6ES7 222-1HH30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	4L
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	DO b.2
3	1L	2L	Kein Anschluss	DO b.3
4	DO a.0	DO a.4	Kein Anschluss	DO b.4
5	DO a.1	DO a.5	3L	DO b.5
6	DO a.2	DO a.6	DO b.0	DO b.6
7	DO a.3	DO a.7	DO b.1	DO b.7

Tabelle A- 97 Anschlussbelegung für das SM 1222 DO 16 x 24 VDC (6ES7 222-1BH30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	DO a.0	DO a.4	DO b.0	DO b.4
5	DO a.1	DO a.5	DO b.1	DO b.5
6	DO a.2	DO a.6	DO b.2	DO b.6
7	DO a.3	DO a.7	DO b.3	DO b.7

### A.6.4 Technische Daten des digitalen Ein-/Ausgangsmoduls SM 1223 (V DC)

Tabelle A- 98 Allgemeine technische Daten

Modell	SM 1223 DI 8 x 24 V DC, DO 8 x Relais	SM 1223 DI 16 x 24 V DC, DO 16 x Relais	SM 1223 DI 8 x 24 V DC, DO 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DO 16 x 24 V DC
Bestellnummer	6ES7 223-1PH30-0XB0	6ES7 223-1PL30-0XB0	6ES7 223-1BH30-0XB0	6ES7 223-1BL30-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75	70 x 100 x 75
Gewicht	230 Gramm	350 Gramm	210 Gramm	310 Gramm
Leistungsverlust	5,5 W	10 W	2,5 W	4,5 W
Stromaufnahme (SM-Bus)	145 mA	180 mA	145 mA	185 mA
Stromaufnahme (24 V DC)	4 mA / Eingang 11 mA / Relaispule		4 mA / Eingang	

Tabelle A- 99 Digitaleingänge

Modell	SM 1223 DI 8 x 24 V DC, DO 8 x Relais	SM 1223 DI 16 x 24 V DC, DO 16 x Relais	SM 1223 DI 8 x 24 V DC, DO 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DO 16 x 24 V DC
Anzahl der Eingänge	8	16	8	16
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)			
Nennspannung	24 V DC bei 4 mA, Nennwert			
Zulässige Dauerspannung	max. 30 V DC			
Stoßspannung	35 V DC für 0,5 s			
Signal logisch 1 (min.)	15 V DC bei 2,5 mA			
Signal logisch 0 (max.)	5 V DC bei 1 mA			
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute			
Potenzialgetrennte Gruppen	2	2	2	2
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 4			



Modell	SM 1223 DI 8 x 24 V DC, DO 8 x Relais	SM 1223 DI 16 x 24 V DC, DO 16 x Relais	SM 1223 DI 8 x 24 V DC, DO 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DO 16 x 24 V DC
Anzahl gleichzeitig eingeschalteter Eingänge	8	<ul style="list-style-type: none"> <li>8 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>16 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	8	16
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt			

Tabelle A- 100 Digitale Ausgänge

Modell	SM 1223 DI 8 x 24 V DC, DO 8 x Relais	SM 1223 DI 16 x 24 V DC, DO 16 x Relais	SM 1223 DI 8 x 24 V DC, DO 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DO 16 x 24 V DC
Ausgänge	8	16	8	16
Typ	Relais, Trockenkontakt		MOSFET, elektronisch (stromliefernd)	
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC		20,4 bis 28,8 V DC	
Signal logisch 1 bei max. Strom	--		min. 20 V DC	
Signal logisch 0 bei 10 kΩ Last	--		max. 0,1 V DC	
Strom (max.)	2,0 A		0,5 A	
Lampenlast	30 W DC/200 W AC		5 W	
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu		max. 0,6 Ω	
Kriechstrom pro Ausgang	--		max. 10 µA	
Einschaltstrom	7 A bei geschlossenen Kontakten		8 A für max. 100 ms	
Überlastschutz	Nein			
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule-Kontakt) Keine (Spule-Logik)		500 V AC für 1 Minute	
Isolationswiderstand	min. 100 MΩ, wenn neu		--	
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute		--	
Potenzialgetrennte Gruppen	2	4	1	1
Strom je Leiter	10 A	8 A	4 A	8 A
Induktive Klemmspannung	--		L+ minus 48 V, 1 W Verlustleistung	
Schaltverzögerung	max. 10 ms		max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus	
Maximale Schaltfrequenz Relais	1 Hz		--	
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu		--	

A.6 Digitale Signalmodule (SM)

Modell	SM 1223 DI 8 x 24 V DC, DO 8 x Relais	SM 1223 DI 16 x 24 V DC, DO 16 x Relais	SM 1223 DI 8 x 24 V DC, DO 8 x 24 V DC	SM 1223 DI 16 x 24 V DC, DO 16 x 24 V DC
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu		--	
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)			
Anzahl gleichzeitig eingeschalteter Ausgänge	8	<ul style="list-style-type: none"> <li>8 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>16 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	8	16
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt			

Tabelle A- 101 Schaltpläne der digitalen Ein-/Ausgangsmodule (V DC/Relais)

SM 1223 DI 8 x 24 VDC, DO 8 x Relais (6ES7 223-1PH30-0XB0)	SM 1223 DI 16 x 24 V DC, DO 16 x Relais (6ES7 223-1PL30-0XB0)	Hinweise
		<p>① Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).</p> <p>Bei stromliefernden Eingängen "+" an "M" anschließen.</p>

Tabelle A- 102 Anschlussbelegung für das SM 1223 DI 8 x 24 V DC, DO 8 x Relais (6ES7 223-1PH30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	GND	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1M	2M	1L	2L
4	DO a.0	DI a.4	DO a.0	DO a.4
5	DI a.1	DI a.5	DO a.1	DO a.5
6	DI a.2	DI a.6	DO a.2	DO a.6
7	DI a.3	DI a.7	DO a.3	DO a.7

Tabelle A- 103 Anschlussbelegung für das SM 1223 DI 16 x 24 V DC, DO 16 x Relais (6ES7 223-1PL30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	GND	1L	3L
2	M / 24 V DC	Kein Anschluss	DO a.0	DO b.0
3	1M	2M	DO a.1	DO b.1
4	DO a.0	DI b.0	DO a.2	DO b.2
5	DI a.1	DI b.1	DO a.3	DO b.3
6	DI a.2	DI b.2	Kein Anschluss	Kein Anschluss
7	DI a.3	DI b.3	2L	4L
8	DI a.4	DI b.4	DO a.4	DO b.4
9	DI a.5	DI b.5	DO a.5	DO b.5
10	DI a.6	DI b.6	DO a.6	DO b.6
11	DI a.7	DI b.7	DO a.7	DO b.7

Tabelle A- 104 Schaltpläne der digitalen Ein-/Ausgangsmodule (V DC)

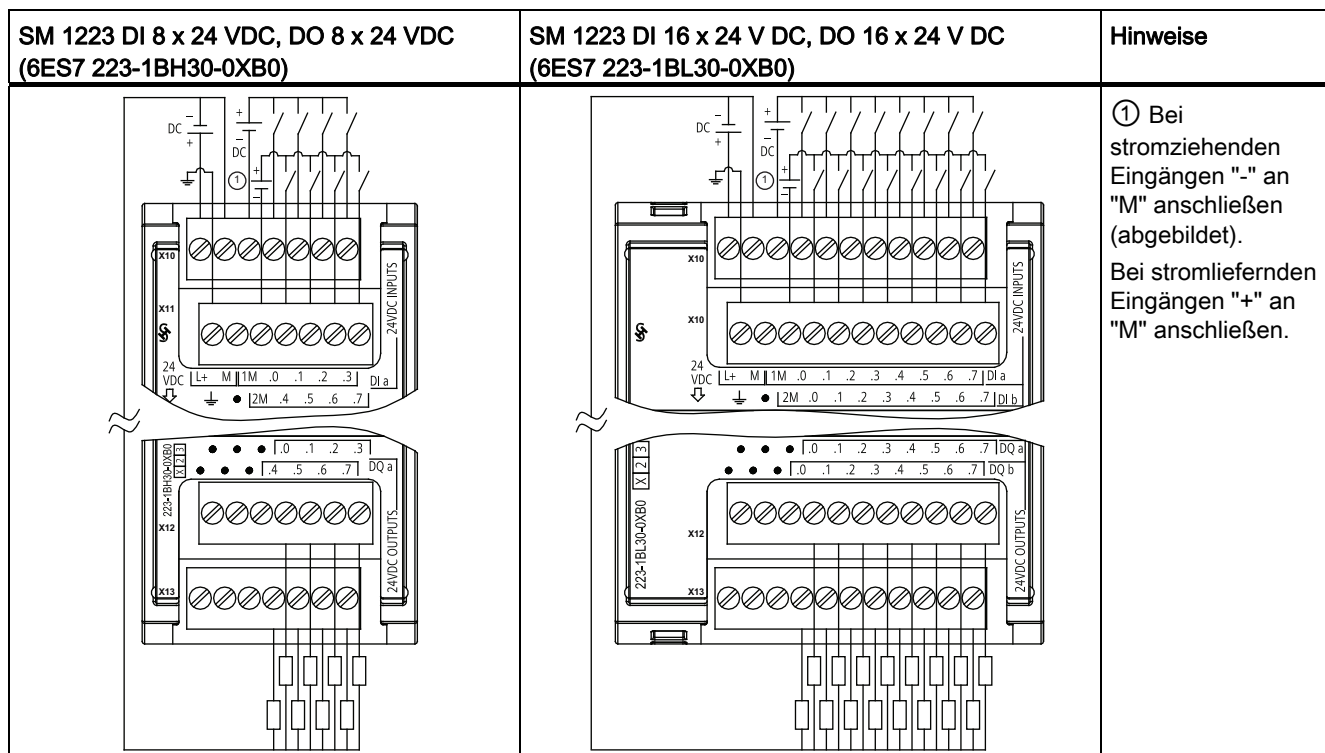


Tabelle A- 105 Anschlussbelegung für das SM 1223 DI 8 x 24 V DC, DO 8 x 24 V DC (6ES7 223-1BH30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	GND	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1M	2M	Kein Anschluss	Kein Anschluss
4	DO a.0	DI a.4	DO a.0	DO a.4
5	DI a.1	DI a.5	DO a.1	DO a.5
6	DI a.2	DI a.6	DO a.2	DO a.6
7	DI a.3	DI a.7	DO a.3	DO a.7

Tabelle A- 106 Anschlussbelegung für das SM 1223 DI 16 x 24 V DC, DO 16 x 24 V DC (6ES7 223-1BL30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	GND	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1M	2M	Kein Anschluss	Kein Anschluss
4	DO a.0	DI b.0	DO a.0	DO b.0
5	DI a.1	DI b.1	DO a.1	DO b.1

Pin	X10	X11	X12	X13
6	DI a.2	DI b.2	DO a.2	DO b.2
7	DI a.3	DI b.3	DO a.3	DO b.3
8	DI a.4	DI b.4	DO a.4	DO b.4
9	DI a.5	DI b.5	DO a.5	DO b.5
10	DI a.6	DI b.6	DO a.6	DO b.6
11	DI a.7	DI b.7	DO a.7	DO b.7

## A.6.5 Technische Daten des digitalen Ein-/Ausgangsmoduls SM 1223 (AC)

Tabelle A- 107 Allgemeine technische Daten

<b>Modell</b>	<b>SM 1223 DI 8 x120/230 V AC / DO 8 x Relais</b>
Bestellnummer	6ES7 223-1QH30-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75 mm
Gewicht	190 Gramm
Leistungsverlust	7,5 W
Stromaufnahme (SM-Bus)	120 mA
Stromaufnahme (24 V DC)	11 mA je Ausgang, wenn eingeschaltet

Tabelle A- 108 Digitaleingänge

<b>Modell</b>	<b>SM 1223 DI 8 x 120/230 V AC / DO 8 x Relais</b>
Anzahl der Eingänge	8
Typ	IEC Typ 1
Nennspannung	120 V AC bei 6 mA, 230 V AC bei 9 mA
Zulässige Dauerspannung	264 V AC
Stoßspannung	--
Signal logisch 1 (min.)	79 V AC bei 2,5 mA
Signal logisch 0 (max.)	20 V AC bei 1 mA
Kriechstrom (max.)	1 mA
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute
Potenzialgetrennte Gruppen <sup>1</sup>	4
Eingabeverzögerungszeiten	Typisch: 0,2 bis 12,8 ms, vom Anwender einstellbar Maximum: -
Anschluss 2Draht-Näherungssensor (Bero) (max.)	1 mA
Leitungslänge	Nicht geschirmt: 300 Meter Geschirmt: 500 Meter
Anzahl gleichzeitig eingeschalteter Eingänge	8

<sup>1</sup> Kanäle in einer Gruppe müssen die gleiche Phase haben.

Tabelle A- 109 Digitale Ausgänge

Modell	SM 1223 DI 8 x 120/230 V AC / DO 8 x Relais
Anzahl der Ausgänge	8
Typ	Relais, Trockenkontakt
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC
Signal logisch 1 bei max. Strom	--
Signal logisch 0 bei 10 kΩ Last	--
Strom (max.)	2,0 A
Lampenlast	30 W DC/200 W AC
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu
Kriechstrom pro Ausgang	--
Einschaltstrom	7 A bei geschlossenen Kontakten
Überlastschutz	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC für 1 Minute (Spule-Kontakt) Keine (Spule-Logik)
Isolationswiderstand	min. 100 MΩ, wenn neu
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute
Potenzialgetrennte Gruppen	2
Strom je Leiter (max.)	10 A
Induktive Klemmspannung	--
Schaltverzögerung (max.)	10 ms
Maximale Schaltfrequenz Relais	1 Hz
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu
Lebensdauer der Kontakte bei Nennlast	1.000.000 Schaltspiele auf/zu
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> <li>• 4 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 8 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt

Tabelle A- 110 SM 1223 DI 8 x 120/230 V AC, DO 8 x Relais (6ES7 223-1QH30-0XB0)

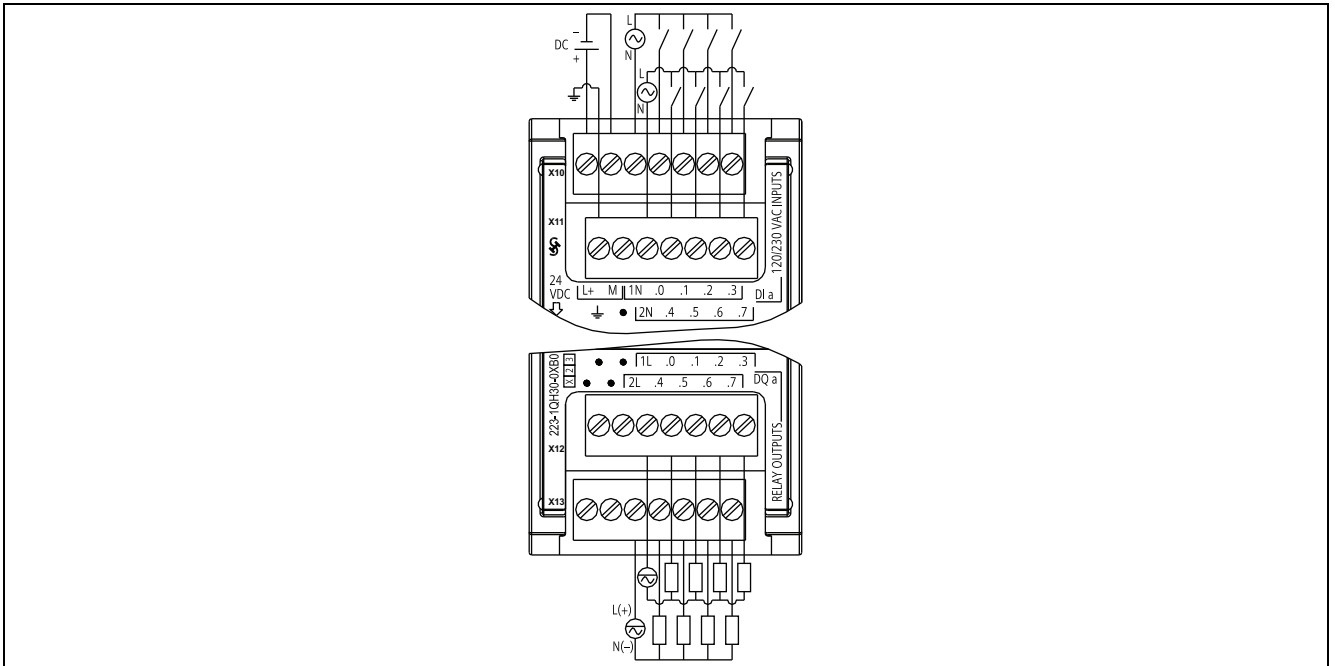


Tabelle A- 111 Anschlussbelegung für das SM 1223 DI 8 x 120/240 VAC, DO 8 x Relais (6ES7 223-1QH30-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	GND	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1N	2N	1L	2L
4	DO a.0	DI a.4	DO a.0	DO a.4
5	DI a.1	DI a.5	DO a.1	DO a.5
6	DI a.2	DI a.6	DO a.2	DO a.6
7	DI a.3	DI a.7	DO a.3	DO a.7

## A.7 Analoge Signalmodule (SM)

### A.7.1 Technische Daten des SM 1231 Analogeingabemoduls

Tabelle A- 112 Allgemeine technische Daten

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Bestellnummer	6ES7 231-4HD30-0XB0	6ES7 231-4HF30-0XB0	6ES7 231-5ND30-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Gewicht	180 Gramm	180 Gramm	180 Gramm
Leistungsverlust	2,2 W	2,3 W	2,0 W
Stromaufnahme (SM-Bus)	80 mA	90 mA	80 mA
Stromaufnahme (24 V DC)	45 mA	45 mA	65 mA

Tabelle A- 113 Analoge Eingänge

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Anzahl der Eingänge	4	8	4
Typ	Spannung oder Strom (differential): wählbar in Gruppen zu je 2		Spannung oder Strom (differential)
Bereich	±10 V, ±5 V, ±2,5 V oder 0 bis 20 mA		±10 V, ±5 V, ±2,5 V, ±1,25 V, 0 bis 20 mA oder 4 mA bis 20 mA
Vollausschlag (Datenwort)	-27.648 bis 27.648 Spannung / 0 bis 27.648 Strom		
Überschwing-/Unterschwingbereich (Datenwort) Siehe Abschnitt über Analogeingangsbereiche für Spannung und Strom (Seite 818).	Spannung: 32.511 bis 27.649 / -27.649 bis -32.512 Strom: 32.511 bis 27.649 / 0 bis -4864		Spannung: 32.511 bis 27.649 / -27.649 bis -32.512 Strom: (0-20 mA): bis 32.511 bis 27.649 / 0 bis -4864; 4-20 mA: 32.511 bis 27.649 / -1 bis -4864
Überlauf/Unterlauf (Datenwort) Siehe Abschnitt über Eingangsbereiche für Spannung und Strom (Seite 818).	Spannung: 32.767 bis 32.512/-32.513 bis -32.768 Strom: 32.767 bis 32.512 / -4865 bis -32.768		Spannung: 32.767 bis 32.512/-32.513 bis -32.768 Strom: 0 bis 20 mA 32.767 bis 32.512 / -4865 bis -32.768 4-20 mA 32.767 bis 32.512/-4865 bis -32.768
Auflösung <sup>1</sup>	12 Bit + Vorzeichenbit		15 Bit + Vorzeichenbit
Maximale Stehspannung/-strom	±35 V / ±40 mA		
Glättung	Keine, schwach, mittel oder stark Siehe Abschnitt zu Schrittantwortzeiten (Seite 817).		
Rauschunterdrückung	400, 60, 50 oder 10 Hz Siehe Abschnitt zu Abtastraten (Seite 817).		



Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Eingangsimpedanz	≥ 9 MΩ (Spannung) / 280 Ω (Strom)		≥ 1 MΩ (Spannung) / < 315 Ω, > 280 Ω (Strom)
Elektrische Trennung Feld zu Logik Logik zu 24 V DC Feld zu 24 V DC Kanal zu Kanal	Keine		500 V AC 500 V AC 500 V AC Keine
Genauigkeit (25 °C / -20 bis 60 °C)	±0,1% / ±0,2% des Vollausschlags		±0,1% / ±0,3% des Vollausschlags
Messprinzip	Istwertumwandlung		
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz		
Betriebssignalbereich <sup>1</sup>	Signal- plus Gleichtaktspannung muss kleiner als +12 V und größer als -12 V sein		
Leitungslänge (Meter)	100 m, verdreht und geschirmt		

<sup>1</sup> Spannungen außerhalb des Betriebsbereichs, die an einem Kanal angelegt werden, können an anderen Kanälen Störungen verursachen.

Tabelle A- 114 Diagnose

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Überlauf/Unterlauf	Ja <sup>1</sup>	Ja	Ja
24 V DC Niederspannung	Ja	Ja	Ja
Drahtbruch	--	--	nur im Bereich 4 bis 20 mA (wenn der Eingang unter - 4164 ist; 1,0 mA)

<sup>1</sup> Bei SM 1231 AI 4 x 13 Bit: Wird am Eingang eine Spannung größer als +30 V DC oder kleiner als -15 V DC angelegt, ist der resultierende Wert unbekannt und der entsprechende Überlauf oder Unterlauf ist möglicherweise nicht aktiv.

Tabelle A- 115 Schaltpläne der Analogeingangs-SMs

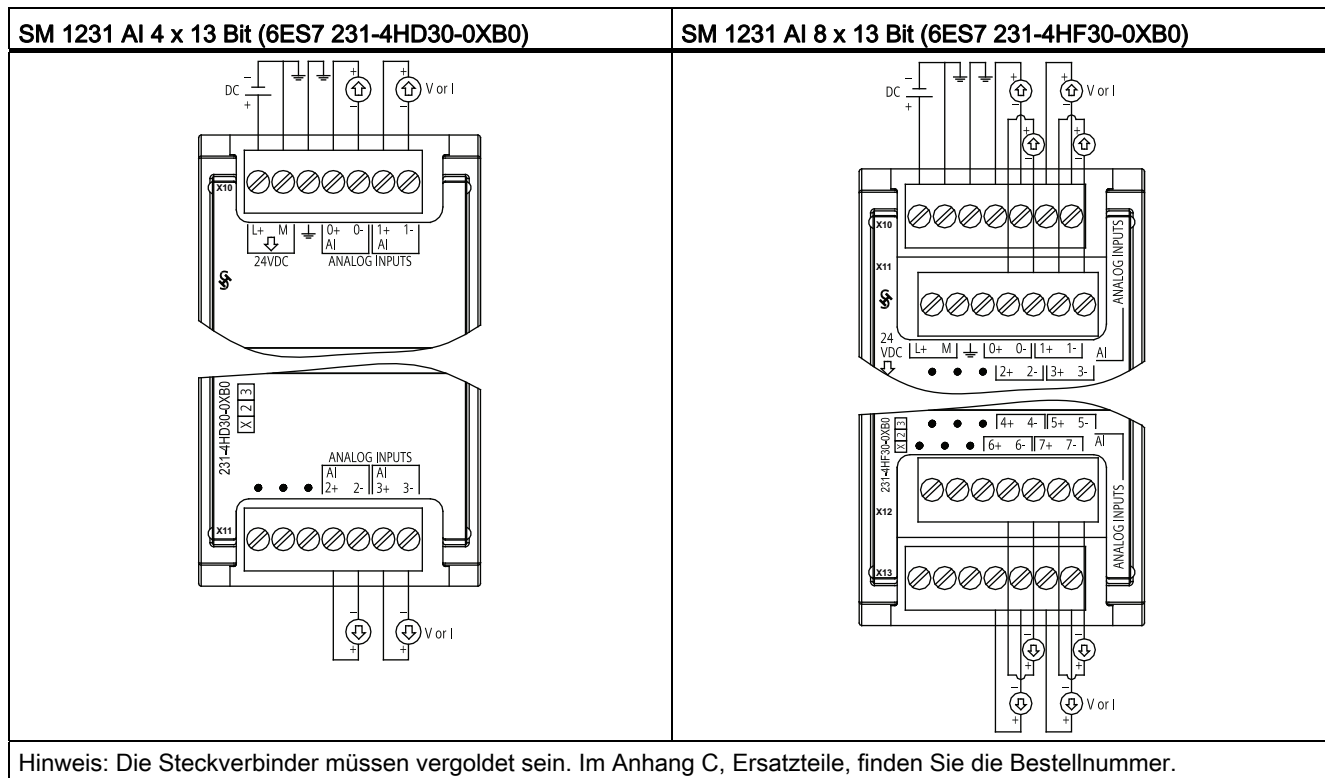


Tabelle A- 116 Anschlussbelegung für das SM 1231 AI 4 x 13 Bit (6ES7 231-4HD30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	GND	Kein Anschluss
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Tabelle A- 117 Anschlussbelegung für das SM 1231 AI 8 x 13 Bit (6ES7 231-4HF30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	GND	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	AI 0+	AI 2+	AI 4+	AI 6+
5	AI 0-	AI 2-	AI 4-	AI 6-
6	AI 1+	AI 3+	AI 5+	AI 7+
7	AI 1-	AI 3-	AI 5-	AI 7-

Tabelle A- 118 Schaltplan des Analogeingangs-SMs

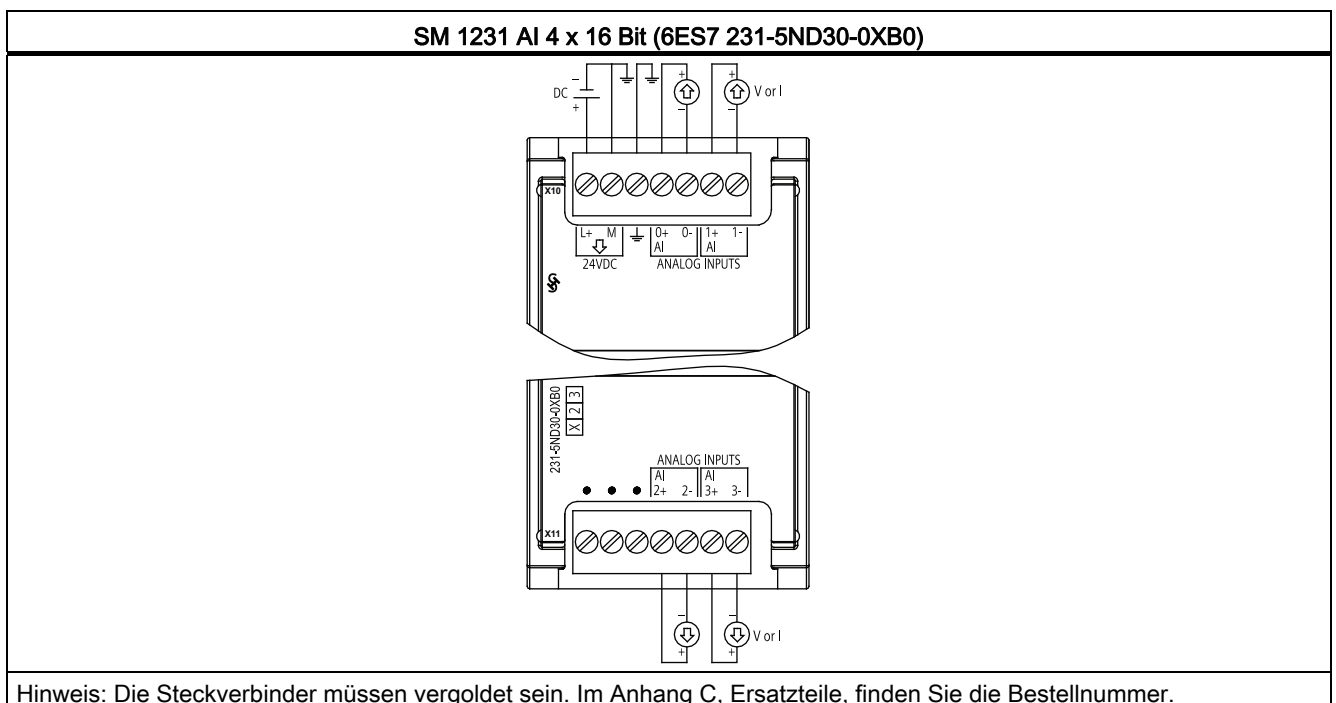


Tabelle A- 119 Anschlussbelegung für das SM 1231 AI 4 x 16 Bit (6ES7 231-5ND30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	GND	Kein Anschluss
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

**Hinweis**

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

Wenn für die Eingänge der Modus "Strom" konfiguriert ist, fließt kein Strom durch den Eingang, es sei denn, Sie versorgen das Modul mit externer Spannung.

**A.7.2 Technische Daten des SM 1232 Analogausgabemoduls**

Tabelle A- 120 Allgemeine technische Daten

Technische Daten	SM 1232 AO 2 x 14 Bit	SM 1232 AO 4 x 14 Bit
Bestellnummer	6ES7 232-4HB30-0XB0	6ES7 232-4HD30-0XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	180 Gramm	180 Gramm
Leistungsverlust	1,8 W	2,0 W
Stromaufnahme (SM-Bus)	80 mA	80 mA
Stromaufnahme (24 V DC)	45 mA (ohne Last)	45 mA (ohne Last)

Tabelle A- 121 Analoge Ausgänge

Technische Daten	SM 1232 AO 2 x 14 Bit	SM 1232 AO 4 x 14 Bit
Ausgänge	2	4
Typ	Spannung oder Strom	Spannung oder Strom
Bereich	±10 V oder 0 bis 20 mA	±10 V oder 0 bis 20 mA
Auflösung	Spannung: 14 Bit Strom: 13 Bit	Spannung: 14 Bit Strom: 13 Bits
Vollausschlag (Datenwort)	Spannung: -27.648 bis 27.648, Strom: 0 bis 27.648 Siehe Abschnitt über Ausgangsbereiche für Spannung und Strom (Seite 819).	
Genauigkeit (25 °C / -20 bis 60 °C)	±0,3 % / ±0,6 % des Vollausschlags	
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 µS (R), 750 µS (1 uF) Strom: 600 µS (1 mH), 2 ms (10 mH)	
Lastimpedanz	Spannung: ≥ 1000 Ω Strom: ≤ 600 Ω	
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Elektrische Trennung (Feld zu Logik)	Keine	
Leitungslänge (Meter)	100 m, verdrillt und geschirmt	

Tabelle A- 122 Diagnose

Technische Daten	SM 1232 AO 2 x 14 Bit	SM 1232 AO 4 x 14 Bit
Überlauf/Unterlauf	Ja	Ja
Erdschluss (nur Spannungsmodus)	Ja	Ja
Drahtbruch (nur Strommodus)	Ja	Ja
24 V DC Niederspannung	Ja	Ja

Tabelle A- 123 Schaltpläne der Analogausgangs-SMs

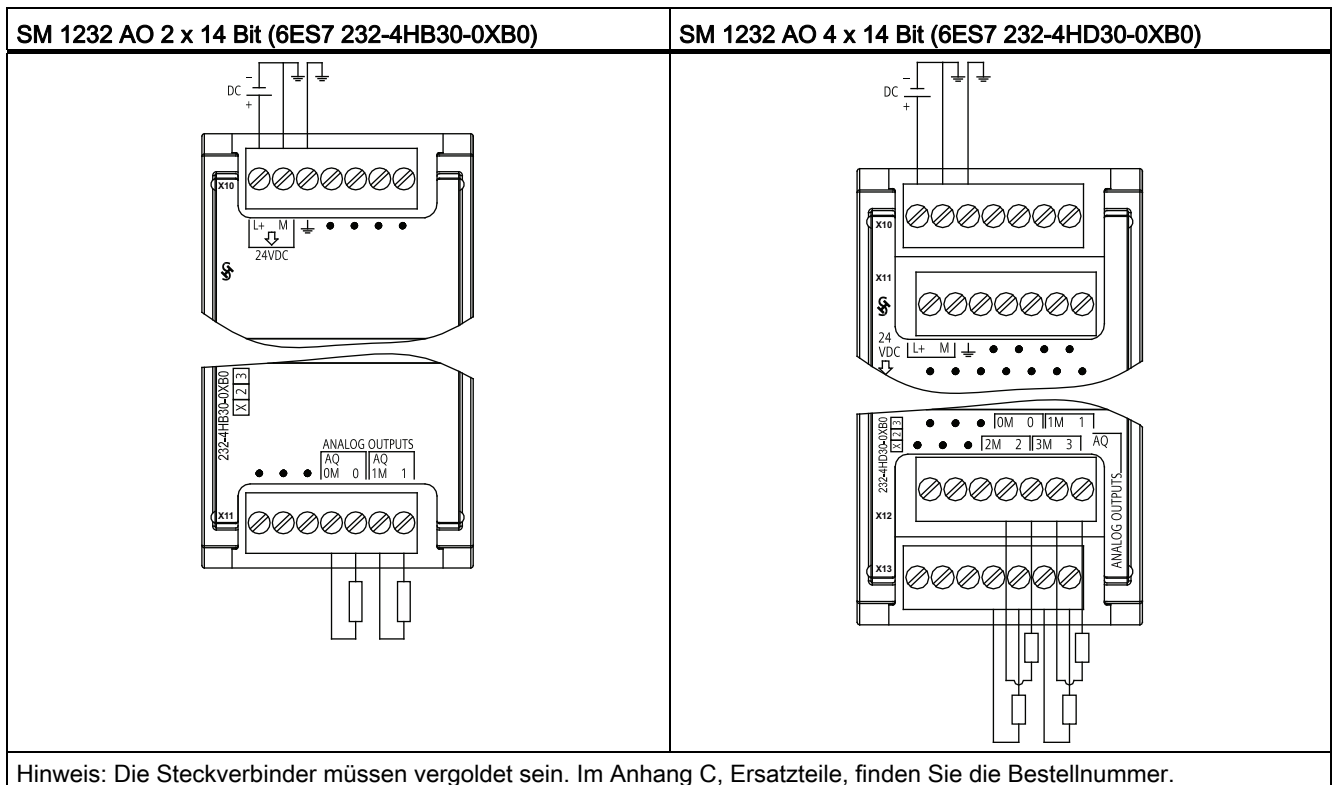


Tabelle A- 124 Anschlussbelegung für das SM 1232 AO 2 x 14 Bit (6ES7 232-4HB30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	GND	Kein Anschluss
4	Kein Anschluss	AO 0M
5	Kein Anschluss	AO 0
6	Kein Anschluss	AO 1M
7	Kein Anschluss	AO 1

Tabelle A- 125 Anschlussbelegung für das SM 1232 AO 4 x 14 Bit (6ES7 232-4HD30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	GND	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	Kein Anschluss	Kein Anschluss	AO 0M	AO 2M
5	Kein Anschluss	Kein Anschluss	AO 0	AO 2
6	Kein Anschluss	Kein Anschluss	AO 1M	AO 3M
7	Kein Anschluss	Kein Anschluss	AO 1	AIO 3

### A.7.3 Technische Daten des SM 1234 Analogein-/Analogausgabemoduls

Tabelle A- 126 Allgemeine technische Daten

Technische Daten	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Bestellnummer	6ES7 234-4HE30-0XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75
Gewicht	220 Gramm
Leistungsverlust	2,4 W
Stromaufnahme (SM-Bus)	80 mA
Stromaufnahme (24 V DC)	60 mA (ohne Last)

Tabelle A- 127 Analoge Eingänge

Modell	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Anzahl der Eingänge	4
Typ	Spannung oder Strom (differential): wählbar in Gruppen zu je 2
Bereich	±10 V, ±5 V, ±2,5 V oder 0 bis 20 mA
Vollausschlag (Datenwort)	-27.648 bis 27.648
Überschwing-/Unterschwingbereich (Datenwort)	Spannung: 32.511 bis 27.649 / -27.649 bis -32.512 Strom: 32.511 bis 27.649 / 0 bis -4864 Siehe Abschnitt über Eingangsbereiche für Spannung und Strom (Seite 818).
Überlauf/Unterlauf (Datenwort)	Spannung: 32.767 bis 32.512 / -32.513 bis -32.768 Strom: 32.767 bis 32.512 / -4865 bis -32.768 Siehe Abschnitt über Eingangsbereiche für Spannung und Strom (Seite 818).
Auflösung	12 Bit + Vorzeichenbit
Maximale Stehspannung/-strom	±35 V / ±40 mA
Glättung	Keine, schwach, mittel oder stark Siehe Abschnitt zu Schrittantwortzeiten (Seite 817).

<b>Modell</b>	<b>SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit</b>
Rauschunterdrückung	400, 60, 50 oder 10 Hz Siehe Abschnitt zu Abtastraten (Seite 817).
Eingangsimpedanz	≥ 9 MΩ (Spannung) / 280 Ω (Strom)
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	±0.1% / ±0.2% des Vollausschlags
Analog-Digital-Umsetzzeit	625 µs (400 Hz Unterdrückung)
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz
Betriebssignalebereich <sup>1</sup>	Signal- plus Gleichtaktspannung muss kleiner als +12 V und größer als -12 V sein
Leitungslänge (Meter)	100 m, verdreht und geschirmt

<sup>1</sup> Spannungen außerhalb des Betriebsbereichs, die an einem Kanal angelegt werden, können an anderen Kanälen Störungen verursachen.

Tabelle A- 128 Analoge Ausgänge

<b>Technische Daten</b>	<b>SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit</b>
Ausgänge	2
Typ	Spannung oder Strom
Bereich	±10 V oder 0 bis 20 mA
Auflösung	Spannung: 14 Bits, Strom: 13 Bits
Vollausschlag (Datenwort)	Spannung: -27.648 bis 27.648, Strom: 0 bis 27.648 Siehe Abschnitt über Ausgangsbereiche für Spannung und Strom (Seite 819).
Genauigkeit (25 °C / -20 bis 60 °C)	±0,3 % / ±0,6 % des Vollausschlags
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 µs (R), 750 µs (1 uF) Strom: 600 µs (1 mH), 2 ms (10 mH)
Lastimpedanz	Spannung: ≥ 1000 Ω Strom: ≤ 600 Ω
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Elektrische Trennung (Feld zu Logik)	Keine
Leitungslänge (Meter)	100 m, verdreht und geschirmt

Tabelle A- 129 Diagnose

<b>Modell</b>	<b>SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit</b>
Überlauf/Unterlauf	Ja <sup>1</sup>
Erdschluss (nur Spannungsmodus)	Ja an Ausgängen
Drahtbruch (nur Strommodus)	Ja an Ausgängen
24 V DC Niederspannung	Ja

<sup>1</sup> Wird am Eingang eine Spannung größer als +30 V DC oder kleiner als -15 V DC angelegt, ist der resultierende Wert unbekannt und der entsprechende Überlauf oder Unterlauf ist möglicherweise nicht aktiv.

Tabelle A- 130 Schaltpläne des analogen Ein-/Ausgangs-SMs

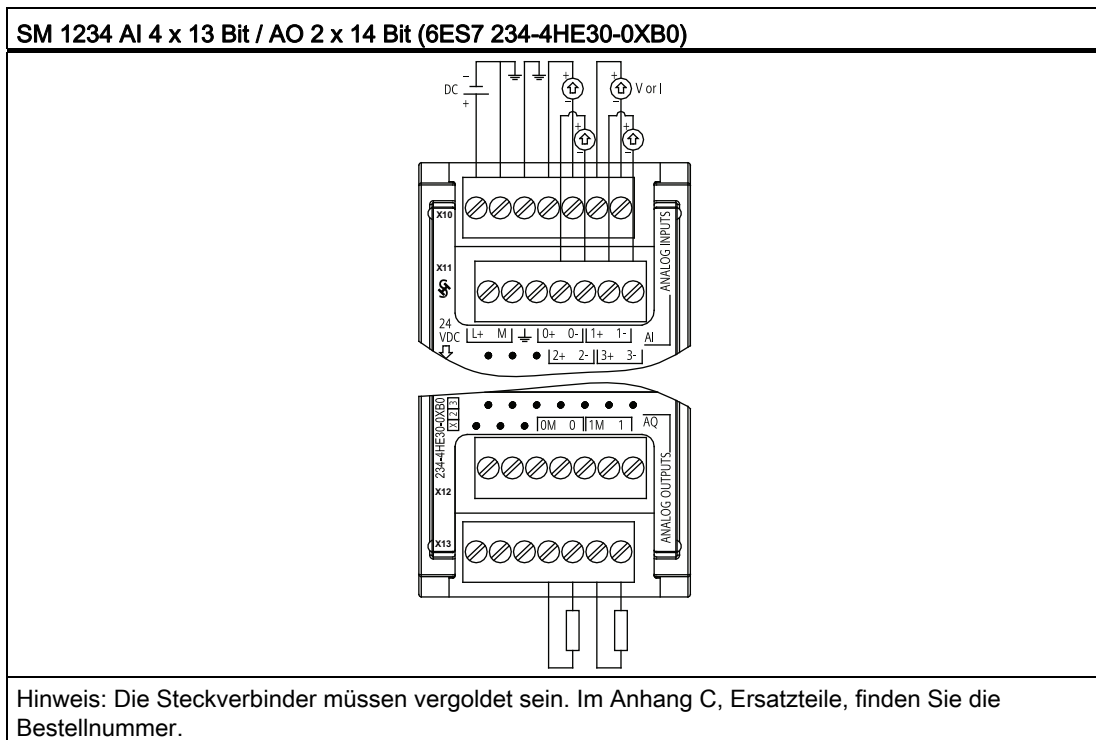


Tabelle A- 131 Anschlussbelegung für das SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit (6ES7 234-4HE30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	GND	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	AI 0+	AI 2+	Kein Anschluss	AO 0M
5	AI 0-	AI 2-	Kein Anschluss	AO 0
6	AI 1+	AI 3+	Kein Anschluss	AO 1M
7	AI 1-	AI 3-	Kein Anschluss	AO 1

**Hinweis**

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

Wenn für die Eingänge der Modus "Strom" konfiguriert ist, fließt kein Strom durch den Eingang, es sei denn, Sie versorgen das Modul mit externer Spannung.



## A.7.4 Schrittantwort der analogen Eingänge

Tabelle A- 132 Schrittantwort (ms), 0 bis Vollausschlag, gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Rauschminderung/Unterdrückungsfrequenz (Auswahl Integrationszeit)			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Keine (1 Zyklus): Keine Mittelwertbildung	4 ms	18 ms	22 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	9 ms	52 ms	63 ms	320 ms
Mittel (16 Zyklen): 16 Abtastwerte	32 ms	203 ms	241 ms	1200 ms
Stark (32 Zyklen): 32 Abtastwerte	61 ms	400 ms	483 ms	2410 ms
<b>Abtastzeit</b>				
• 4 AI x 13 Bit	• 0,625 ms	• 4,17 ms	• 5 ms	• 25 ms
• 8 AI x 13 Bit	• 1,25 ms	• 4,17 ms	• 5 ms	• 25 ms
• 4 AI4 x 16 Bit	• 0,417 ms	• 0,397 ms	• 0,400 ms	• 0,400 ms

## A.7.5 Abtastzeit und Aktualisierungszeiten der Analogeingänge

Tabelle A- 133 Abtastzeit und Aktualisierungszeit

Unterdrückungsfrequenz (Integrationszeit)	Abtastzeit	Aktualisierungszeit des Moduls für alle Kanäle	
		4-kanaliges SM	8-kanaliges SM
400 Hz (2,5 ms)	<ul style="list-style-type: none"> <li>• 4-kanaliges SM: 0,625 ms</li> <li>• 8-kanaliges SM: 1,250 ms</li> </ul>	0,625 ms	1,250 ms
60 Hz (16,6 ms)	4,170 ms	4,17 ms	4,17 ms
50 Hz (20 ms)	5,000 ms	5 ms	5 ms
10 Hz (100 ms)	25,000 ms	25 ms	25 ms

### A.7.6 Messbereiche der analogen Eingänge für Spannung

Tabelle A- 134 Darstellung Analogeingang für Spannung

System		Messbereich Spannung							
Dezimal	Hexadezimal	±10 V	±5 V	±2,5 V	±1,25 V	0 bis 10 V			
32767	7FFF	11,851 V	5,926 V	2,963 V	1,481 V	Überlauf	11,851 V	Überlauf	
32512	7F00								
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Überschwingbereich	11,759 V	Überschwingbereich	
27649	6C01								
27648	6C00	10 V	5 V	2,5 V	1,250 V	Bemessungsbereich	10 V	Bemessungsbereich	
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V		7,5 V		
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV		361,7 µV		
0	0	0 V	0 V	0 V	0 V		0 V		
-1	FFFF						Negative Werte werden nicht unterstützt		
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V				
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V				
-27649	93FF					Unterschwingbereich			
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V				
-32513	80FF					Unterlauf			
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V				

### A.7.7 Messbereiche der analogen Eingänge für Strom

Tabelle A- 135 Darstellung Analogeingang für Strom

System		Messbereich Strom			
Dezimal	Hexidezimal	0 mA bis 20 mA	4 mA bis 20 mA		
32767	7FFF	23,70 mA	22,96 mA	Überlauf	
32512	7F00				
32511	7EFF	23,52 mA	22,81 mA	Überschwingbereich	
27649	6C01				
27648	6C00	20 mA	20 mA	Nennbereich	
20736	5100	15 mA	16 mA		
1	1	723,4 nA	4 mA + 578,7 nA		
0	0	0 mA	4 mA		
-1	FFFF				Unterschwingbereich
-4864	ED00	-3,52 mA	1,185 mA		
-4865	ECFF			Unterlauf	
-32768	8000				

## A.7.8 Ausgangsmessbereiche (AO) für Spannung und Strom (SB und SM)

Tabelle A- 136 Darstellung Analogausgang für Spannung

System		Spannungsausgangsbereich	
Dezimal	Hexadezimal	±10 V	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	
32511	7EFF	11,76 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11,76 V	Unterschwingbereich
-32513	80FF	Siehe Hinweis 1	
-32768	8000	Siehe Hinweis 1	
			Unterlauf

<sup>1</sup> Bei Überlauf oder Unterlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration des Analogsignalmoduls. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: Ersatzwert aufschalten oder Letzten Wert halten.

Tabelle A- 137 Darstellung Analogausgang für Strom

System		Stromausgangsbereich	
Dezimal	Hexadezimal	0 mA bis 20 mA	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	
32511	7EFF	23,52 mA	Überschwingbereich
27649	6C01		
27648	6C00	20 mA	Bemessungsbereich
20736	5100	15 mA	
1	1	723,4 nA	
0	0	0 mA	

<sup>1</sup> Bei Überlauf oder Unterlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration des Analogsignalmoduls. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: Ersatzwert aufschalten oder Letzten Wert halten.

## A.8 Thermoelement- und RTD-Signalmodule (SMs)

### A.8.1 SM 1231 Thermoelement

Tabelle A- 138 Allgemeine technische Daten

Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC
Bestellnummer	6ES7 231-5QD30-0XB0	6ES7 231-5QF30-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	180 Gramm	190 Gramm
Leistungsverlust	1,5 W	1,5 W
Stromaufnahme (SM-Bus)	80 mA	80 mA
Stromaufnahme (24 V DC) <sup>1</sup>	40 mA	40 mA

<sup>1</sup> 20,4 bis 28,8 V DC (Klasse 2, leistungsbegrenzt oder Geberspannung aus SPS)

Tabelle A- 139 Analoge Eingänge

Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC	
Anzahl der Eingänge	4	8	
Bereich	Siehe Thermoelement-Auswahltabelle (Seite 824).	Siehe Thermoelement-Auswahltabelle (Seite 824).	
Nennbereich (Datenwort)			
Überlauf/Unterlauf (Datenwort)			
Überlauf/Unterlauf (Datenwort)			
Auflösung	Temperaturbereich	0,1 °C/0,1 °F	0,1 °C/0,1 °F
	Spannung	15 Bit plus Vorzeichen	15 Bit plus Vorzeichen
Max. Stehspannung	±35 V	±35 V	
Rauschunterdrückung	85 dB für die gewählte Filtereinstellung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)	85 dB für die gewählte Filtereinstellung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)	
Gleichtaktunterdrückung	> 120 dB bei 120 V AC	> 120 dB bei 120 V AC	
Impedanz	≥ 10 MΩ	≥ 10 MΩ	
Elektrische Trennung	Feld zu Logik	500 V AC	500 V AC
	Feld zu 24 V DC	500 V AC	500 V AC
	24 V DC zu Logik	500 V AC	500 V AC
Kanal zu Kanal	120 V AC	120 V AC	
Genauigkeit	Siehe Thermoelement-Auswahltabelle (Seite 824).	Siehe Thermoelement-Auswahltabelle (Seite 824).	
Wiederholgenauigkeit	±0,05 % Vollausschlag	±0,05 % Vollausschlag	
Messprinzip	Integrierend	Integrierend	
Aktualisierungszeit Modul	Siehe Auswahltabelle Rauschminderung (Seite 824).	Siehe Auswahltabelle Rauschminderung (Seite 824).	

Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC
Fehler kalte Verbindungsstelle	±1.5°C	±1.5°C
Leitungslänge (Meter)	Max. 100 m zum Geber	Max. 100 m zum Geber
Leitungswiderstand	max. 100 Ω	max. 100 Ω

Tabelle A- 140 Diagnose

Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC
Überlauf/Unterlauf <sup>1</sup>	Ja	Ja
Drahtbruch (nur Strommodus) <sup>2</sup>	Ja	Ja
24-V-DC-Niederspannung <sup>1</sup>	Ja	Ja

<sup>1</sup> Die Informationen der Diagnosealarme "Überlauf", "Unterlauf" und "Niederspannung" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.

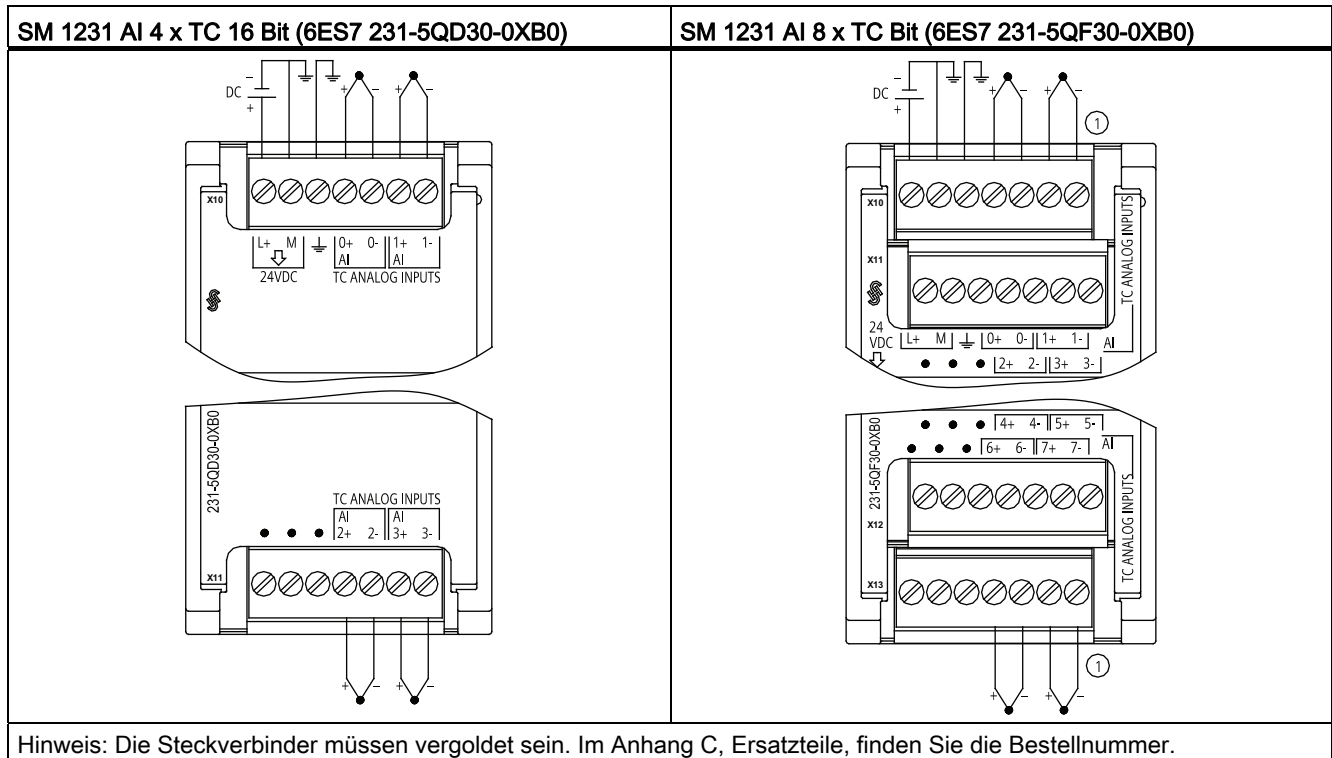
<sup>2</sup> Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Das analoge Thermoelement-Signalmodul SM 1231 TC misst den Wert der an die Moduleingänge angeschlossenen Spannung. Als Temperaturmessart sind entweder "Thermoelement" oder "Spannung" möglich.

- "Thermoelement": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).

- "Spannung": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.

Tabelle A- 141 Schaltpläne der Thermoelement-SMs



① Die Anschlüsse von TC 2, 3, 4 und 5 werden zur besseren Übersichtlichkeit weggelassen.

Tabelle A- 142 Anschlussbelegung für das SM 1231 AI 4 x TC 16 Bit (6ES7 231-5QD30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	GND	Kein Anschluss
4	AI 0+ /TC	AI 2+ /TC
5	AI 0- /TC	AI 2- /TC
6	AI 1+ /TC	AI 3+ /TC
7	AI 1- /TC	AI 3- /TC

Tabelle A- 143 Anschlussbelegung für das SM 1231 AI 8 x TC Bit (6ES7 231-5QF30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	GND	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	AI 0+ /TC	AI 2+ /TC	AI 4 I- /TC	AI 6 I- /TC
5	AI 0- /TC	AI 2- /TC	AI 4 I+ /TC	AI 6 I+ /TC
6	AI 1+ /TC	AI 3+ /TC	AI 5 M- /TC	AI 7 M- /TC
7	AI 1- /TC	AI 3- /TC	AI 5 M+ /TC	AI 7 M+ /TC

**Hinweis**

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

Die vom Thermoelement nicht verwendeten Kanäle können deaktiviert werden. Es tritt kein Fehler auf, wenn ein nicht verwendeter Kanal deaktiviert wird.

**A.8.1.1****Grundlegende Funktionsweise eines Thermoelements**

Thermoelemente entstehen, wenn zwei unterschiedliche Metalle elektrisch miteinander verbunden werden. Dadurch wird eine Spannung erzeugt, die proportional zu der Temperatur der Verbindungsstelle ist. Es handelt sich um eine geringe Spannung. Ein Mikrovolt kann viele Grade darstellen. Grundlage für die Temperaturmessung mit Thermoelementen sind das Messen der Spannung eines Thermoelements, das Kompensieren von zusätzlichen Verbindungsstellen und das Linearisieren der Ergebnisse.

Wenn Sie ein Thermoelement an das SM 1231 Thermoelementmodul anschließen, werden die beiden Leitungen der unterschiedlichen Metalle am Signalanschluss des Moduls angeschlossen. Die Stelle, an der die beiden unterschiedlichen Leitungen miteinander verbunden werden, bildet den Sensor des Thermoelements.

Zwei weitere Thermoelemente entstehen an der Stelle, an der die unterschiedlichen Leitungen an den Signalanschluss angeschlossen werden. Die Temperatur des Anschlusses erzeugt eine Spannung, die zu der Spannung des Thermoelementsensors addiert wird. Wird diese Spannung nicht ausgeglichen, weicht die ausgegebene Temperatur von der Temperatur des Sensors ab.

Durch die Kompensation der kalten Verbindungsstelle wird das Thermoelement am Anschluss ausgeglichen. Tabellen für Thermoelemente basieren auf einer Bezugstemperatur an der Verbindungsstelle, üblicherweise Null Grad Celsius. Durch die Kompensation der kalten Verbindungsstelle wird der Anschluss auf Null Grad Celsius kompensiert. Die Spannung, die durch das Thermoelement des Anschlusses addiert wird, wird durch die Kompensation der kalten Verbindungsstelle korrigiert. Die Temperatur des Moduls wird innen gemessen und dann in einen Wert umgewandelt, der zur Sensorumsetzung addiert wird. Die korrigierte Sensorumsetzung wird dann mittels der Thermoelement-Tabellen linearisiert.

A.8 Thermoelement- und RTD-Signalmodule (SMs)

Für die optimale Funktionsweise der Kompensation der kalten Verbindungsstelle muss sich das Thermoelementmodul in einer thermisch stabilen Umgebung befinden. Ein langsame Veränderung (weniger als 0,1 °C/Minute) der Temperatur in der Umgebung des Moduls wird innerhalb der Modulspezifikation korrekt ausgeglichen. Auch Luftbewegungen am Modul verursachen Fehler bei der Kompensation der kalten Verbindungsstelle.

Ist eine bessere Kompensation von Fehlern an der kalten Verbindungsstelle erforderlich, kann eine externe isothermische Klemmenleiste verwendet werden. Mit dem Thermoelementmodul kann eine auf 0 °C bezogene oder eine auf 50 °C bezogene Klemmenleiste eingesetzt werden.

**A.8.1.2 Auswahltabellen für das SM 1231 Thermoelement**

Die Bereiche und Genauigkeit der verschiedenen vom SM 1231 Thermoelement-Signalmodul unterstützten Thermoelementtypen entnehmen Sie bitte der unten stehenden Tabelle.

Tabelle A- 144 SM 1231 Thermoelement-Auswahltabelle

Typ	Minimum unterer Bereich <sup>1</sup>	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich <sup>2</sup>	Normalbereich <sup>3,4</sup> Genauigkeit bei 25 °C	Normalbereich <sup>1,2</sup> Genauigkeit -20 °C bis 60 °C
J	-210,0 °C	-150,0 °C	1200,0 °C	1450,0 °C	±0,3 °C	±0,6 °C
K	-270,0 °C	-200,0 °C	1372,0 °C	1622,0 °C	±0,4 °C	±1,0 °C
T	-270,0 °C	-200,0 °C	400,0 °C	540,0 °C	±0,5 °C	±1,0 °C
E	-270,0 °C	-200,0 °C	1000,0 °C	1200,0 °C	±0,3 °C	±0,6 °C
R & S	-50,0 °C	100,0 °C	1768,0 °C	2019,0 °C	±1,0 °C	±2,5 °C
N	-270,0 °C	-200,0 °C	1300,0 °C	1550,0 °C	±1,0 °C	±1,6 °C
C	0,0 °C	100,0 °C	2315,0 °C	2500,0 °C	±0,7 °C	±2,7 °C
TXK/XK(L)	-200,0 °C	-150,0 °C	800,0 °C	1050,0 °C	±0,6 °C	±1,2 °C
Spannung	-32512	-27648 -80 mV	27648 80 mV	32511	±0.05%	±0.1%

- 1 Die Thermoelementwerte unterhalb des Minimums für den unteren Bereich werden als -32768 angegeben.
- 2 Die Thermoelementwerte oberhalb des Minimums für den oberen Bereich werden als 32767 angegeben.
- 3 Der interne Fehler an der kalten Verbindungsstelle beträgt ±1,5 °C für alle Bereiche. Dieser Wert ist zum in dieser Tabelle aufgeführten Fehler zu addieren. Das Modul benötigt eine Aufwärmzeit von mindestens 30 Minuten, bis die hier genannten Spezifikationen erfüllt werden.
- 4 Bei Vorhandensein von abgestrahlter Funkfrequenz zwischen 970 MHz und 990 MHz kann sich die Genauigkeit des SM 1231 AI 4 x 16 Bit TC verschlechtern.

Tabelle A- 145 Rauschminderung und Aktualisierungszeiten für das SM 1231 Thermoelement

Auswahl Unterdrückungsfrequenz	Integrationszeit	Aktualisierungszeit 4-Kanal-Modul(Sekunden)	Aktualisierungszeit 8-Kanal-Modul(Sekunden)
400 Hz (2,5 ms)	10 ms <sup>1</sup>	0.143	0.285
60 Hz (16,6 ms)	16,67 ms	0.223	0.445



Auswahl Unterdrückungsfrequenz	Integrationszeit	Aktualisierungszeit 4-Kanal- Modul(Sekunden)	Aktualisierungszeit 8-Kanal- Modul(Sekunden)
50 Hz (20 ms)	20 ms	0.263	0.525
10 Hz (100 ms)	100 ms	1.225	2.450

- <sup>1</sup> Um die Auflösung und Messgenauigkeit des Moduls bei Auswahl der 400-Hz-Unterdrückung aufrecht zu erhalten, beträgt die Integrationszeit 10 ms. Mit dieser Auswahl erfolgt auch die Rauschunterdrückung bei 100 Hz und 200 Hz.

Für die Messung von Thermoelementen wird eine Integrationszeit von 100 ms empfohlen. Niedriger eingestellte Integrationszeiten führen zu einem höheren Wiederholgenauigkeitsfehler der Temperaturmessungen.

### Hinweis

Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm ggf. berücksichtigt werden. Weil sich die Konfiguration des Moduls auf die Dauer der Initialisierungszeit auswirken kann, prüfen Sie das Verhalten des Moduls in Ihrer Konfiguration. Ggf. können Sie Logik in Ihr Anwenderprogramm aufnehmen, um die Initialisierungszeit des Moduls unterzubringen.

## Darstellung der Analogwerte für ein Thermoelement vom Typ J

Die Analogwerte für ein Thermoelement vom Typ J finden Sie in der folgenden Tabelle.

Tabelle A- 146 Darstellung der Analogwerte für ein Thermoelement vom Typ J

Typ J in °C	Einheiten		Typ J in °F	Einheiten		Bereich
	Dezimal	Hexadezimal		Dezimal	Hexadezimal	
> 1450.0	32767	7FFF	> 2642.0	32767	7FFF	Überlauf
1450.0	14500	38A4	2642.0	26420	6734	Oberhalb des Bereichs
:	:	:	:	:	:	
1200.1	12001	2EE1	2192.2	21922	55A2	Bemessungs bereich
1200.0	12000	2EE0	2192.0	21920	55A0	
:	:	:	:	:	:	Unterlauf <sup>1</sup>
-150.0	-1500	FA24	-238.0	-2380	F6B4	
< -150.0	-32768	8000	< -238.0	-32768	8000	

<sup>1</sup> Fehlerhafte Verdrahtung (z. B. Verpolung oder offene Eingänge) oder ein Sensorfehler im negativen Bereich (z. B. falscher Typ des Thermoelements) können bewirken, dass das Thermoelementmodul Unterlauf meldet.

## A.8.2 SM 1231 RTD

### Technische Daten des SM 1231 RTD

Tabelle A- 147 Allgemeine technische Daten

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x 16 Bit
Bestellnummer	6ES7 231-5PD30-0XB0	6ES7 231-5PF30-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	70 x 100 x 75
Gewicht	220 Gramm	270 Gramm
Leistungsverlust	1,5 W	1,5 W
Stromaufnahme (SM-Bus)	80 mA	90 mA
Stromaufnahme (24 V DC) <sup>1</sup>	40 mA	40 mA

<sup>1</sup> 20,4 bis 28,8 V DC (Klasse 2, leistungsbegrenzt oder Geberspannung aus CPU)

Tabelle A- 148 Analoge Eingänge

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x16 Bit
Anzahl der Eingänge	4	8
Typ	Modulreferenz RTD und $\Omega$	Modulreferenz RTD und $\Omega$
Bereich	Siehe Auswahltablette RTD-Geber	Siehe Auswahltablette RTD-Geber
Nennbereich (Datenwort)	(Seite 829).	(Seite 829).
Überschwing-/Unterschwingbereich (Datenwort)		
Überlauf/Unterlauf (Datenwort)		
Auflösung		
Temperaturbereich	0,1 °C/0,1 °F	0,1 °C/0,1 °F
Widerstand	15 Bit plus Vorzeichen	15 Bit plus Vorzeichen
Max. Stehspannung	±35 V	±35 V
Rauschunterdrückung	85 dB für die gewählte Rauschminderung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)	85 dB für die gewählte Rauschminderung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)
Gleichtaktunterdrückung	>120 dB	>120 dB
Impedanz	≥ 10 M $\Omega$	≥ 10 M $\Omega$
Elektrische Trennung		
Feldseite zu Logik	500 V AC	500 V AC
Feld zu 24 V DC	500 V AC	500 V AC
24 V DC zu Logik	500 V AC	500 V AC
Trennung Kanäle untereinander	Keine	Keine
Genauigkeit	Siehe Auswahltablette RTD-Geber (Seite 829).	Siehe Auswahltablette RTD-Geber (Seite 829).
Wiederholgenauigkeit	±0,05 % Vollausschlag	±0,05 % Vollausschlag
Maximale Verlustleistung Geber	0,5 m W	0,5 m W
Messprinzip	Integrierend	Integrierend

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x16 Bit
Aktualisierungszeit Modul	Siehe Auswahltabelle Rauschminderung (Seite 829).	Siehe Auswahltabelle Rauschminderung (Seite 829).
Leitungslänge (Meter)	Max. 100 m zum Geber	Max. 100 m zum Geber
Leitungswiderstand	20 $\Omega$ , 2,7 $\Omega$ für max. 10 $\Omega$ RTD	20 $\Omega$ , 2,7 $\Omega$ für max. 10 $\Omega$ RTD

Tabelle A- 149 Diagnose

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x16 Bit
Überlauf/Unterlauf <sup>1,2</sup>	Ja	Ja
Drahtbruch <sup>3</sup>	Ja	Ja
24-V-DC-Niederspannung <sup>1</sup>	Ja	Ja

<sup>1</sup> Die Informationen der Diagnosealarme "Überlauf", "Unterlauf" und "Niederspannung" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.

<sup>2</sup> Für Widerstandsmessbereiche ist die Prüfung auf Unterlauf grundsätzlich nicht aktiviert.

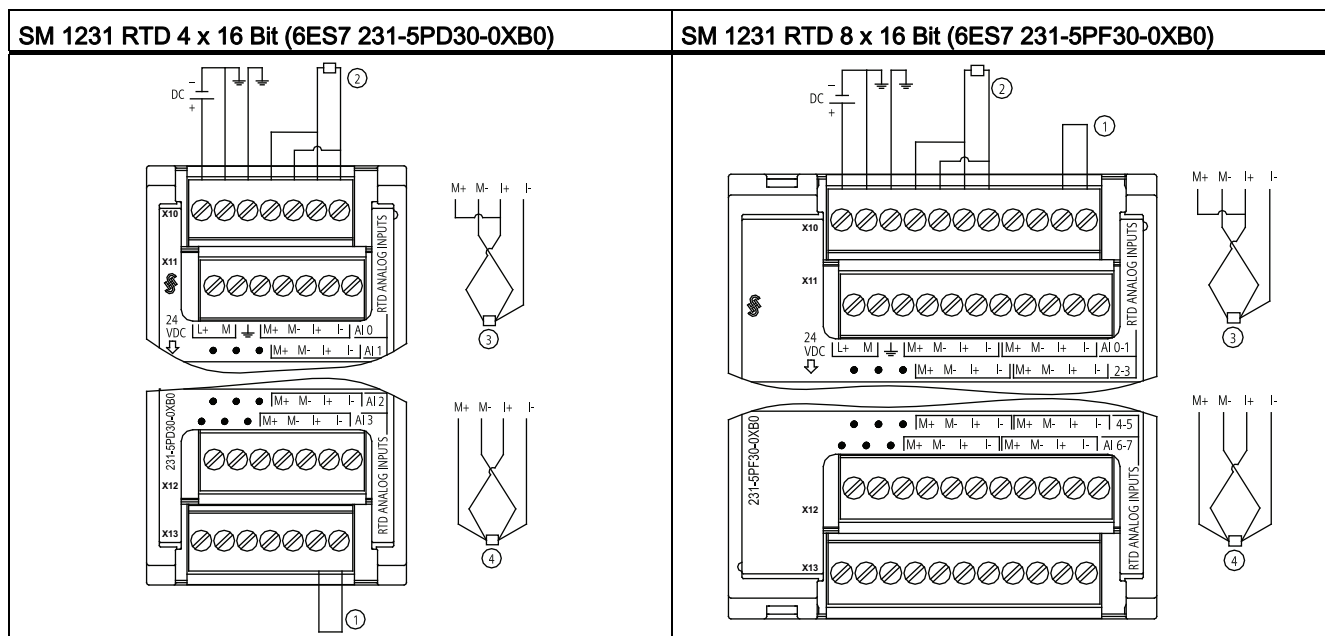
<sup>3</sup> Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Das analoge Signalmodul SM 1231 RTD misst den Wert des an die Moduleingänge angeschlossenen Widerstands. Als Messart kann entweder "Widerstand" oder "Thermischer Widerstand" ausgewählt werden.

- "Widerstand": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.
- "Thermischer Widerstand": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).

Das Signalmodul SM 1231 RTD unterstützt Messungen über 2-Leiter-, 3-Leiter- und 4-Leiter-Anschlüsse zum Geberwiderstand.

Tabelle A- 150 Schaltpläne der RTD-SMs



- ① Nicht belegte RTD-Eingänge zurückschleifen
- ② 2-Draht-RTD ③ 3-Draht-RTD ④ 4-Draht-RTD

HINWEIS: Hinweis: Die Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 151 Anschlussbelegung für das SM 1231 RTD 4 x 16 Bit (6ES7 231-5PD30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	GND	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	AI 0 M+ /RTD	AI 1 M+ /RTD	AI 2 M+ /RTD	AI 3 M+ /RTD
5	AI 0 M- /RTD	AI 1 M- /RTD	AI 2 M- /RTD	AI 3 M- /RTD
6	AI 0 I+ /RTD	AI 1 I+ /RTD	AI 2 I+ /RTD	AI 3 I+ /RTD
7	AI 0 I- /RTD	AI 1 I- /RTD	AI 2 I- /RTD	AI 3 I- /RTD

Tabelle A- 152 Anschlussbelegung für das SM 1231 RTD 8 x 16 Bit (6ES7 231-5PF30-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	GND	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	AI 0 M+ /RTD	AI 2 M+ /RTD	AI 4 M+ /RTD	AI 6 M+ /RTD
5	AI 0 M- /RTD	AI 2 M- /RTD	AI 4 M- /RTD	AI 6 M- /RTD
6	AI 0 I+ /RTD	AI 2 I+ /RTD	AI 4 I+ /RTD	AI 6 I+ /RTD

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
7	AI 0 I- /RTD	AI 2 I- /RTD	AI 4 I- /RTD	AI 6 I- /RTD
8	AI 1 M+ /RTD	AI 3 M+ /RTD	AI 5 M+ /RTD	AI 7 M+ /RTD
9	AI 1 M- /RTD	AI 3 M- /RTD	AI 5 M- /RTD	AI 7 M- /RTD
10	AI 1 I+ /RTD	AI 3 I+ /RTD	AI 5 I+ /RTD	AI 7 I+ /RTD
11	AI 1 I- /RTD	AI 3 I- /RTD	AI 5 I- /RTD	AI 7 I- /RTD

**Hinweis**

Die vom RTD nicht verwendeten Kanäle können deaktiviert werden. Es tritt kein Fehler auf, wenn ein nicht verwendeter Kanal deaktiviert wird.

Das RTD-Modul benötigt eine kontinuierliche Stromschleife, um die zusätzliche Stabilisierungszeit zu beseitigen, die bei einem nicht verwendeten Kanal, der nicht deaktiviert ist, automatisch anfällt. Aus Konsistenzgründen muss ein Widerstand an das RTD-Modul angeschlossen werden (z. B. der 2-Draht-RTD-Anschluss).

**A.8.2.1 Auswahltabellen für das SM 1231 RTD**

Tabelle A- 153 Bereiche und Genauigkeit für die verschiedenen Geber, die von den RTD-Modulen unterstützt werden

Temperaturkoeffizient	RTD-Typ	Minimum unterer Bereich <sup>1</sup>	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich <sup>2</sup>	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich bei -20 °C bis 60 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 10	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±1,0 °C	±2,0 °C
	Pt 50	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003902 Pt 0,003916 Pt 0,003920	Pt 100	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 200	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±1,0 °C	±2,0 °C
	Pt 50	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±0,8 °C	±1,6 °C
	Pt 100						
	Pt 500						
Ni 0,006720 Ni 0,006180	Ni 100	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
	Ni 120						
	Ni 200						
	Ni 500						

Temperaturkoeffizient	RTD-Typ	Minimum unterer Bereich <sup>1</sup>	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich <sup>2</sup>	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
	Ni 1000						
LG-Ni 0,005000	LG-Ni 1000	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
Ni 0,006170	Ni 100	-105,0 °C	-60,0 °C	180,0 °C	212,4 °C	±0,5 °C	±1,0 °C
Cu 0,004270	Cu 10	-240,0 °C	-200,0 °C	260,0 °C	312,0 °C	±1,0 °C	±2,0 °C
Cu 0,004260	Cu 10	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±0,6 °C	±1,2 °C
	Cu 100						
Cu 0,004280	Cu 10	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±0,7 °C	±1,4 °C
	Cu 100						

<sup>1</sup> Die RTD-Werte unterhalb des Minimums für den unteren Bereich werden als -32768 ausgegeben.

<sup>2</sup> RTD-Werte oberhalb des Maximums für den oberen Bereich werden als +32767 ausgegeben.

Tabelle A- 154 Widerstand

Bereich	Minimum unterer Bereich	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich <sup>1</sup>	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
150 Ω	nicht zutreffend	0 (0 Ω)	27648 (150 Ω)	176.383 Ω	±0.05%	±0.1%
300 Ω	nicht zutreffend	0 (0 Ω)	27648 (300 Ω)	352.767 Ω	±0.05%	±0.1%
600 Ω	nicht zutreffend	0 (0 Ω)	27648 (600 Ω)	705.534 Ω	±0.05%	±0.1%

<sup>1</sup> Die Widerstandswerte oberhalb des Minimums für den oberen Bereich werden als +32767 ausgegeben.

**Hinweis**

Für alle aktivierte Kanäle ohne angeschlossenen Geber meldet das Modul 32767. Wenn außerdem die Prüfung auf offene Leitungen aktiviert ist, blinken am Modul die entsprechenden roten LED.

Wenn RTD-Bereiche von 500 Ω und 1000 Ω mit anderen Widerständen niederen Werts verwendet werden, kann sich der Fehler auf den zweifachen spezifizierten Fehler erhöhen.

Optimale Genauigkeit für die 10 Ω-RTD-Bereiche ermöglichen 4-Leiter-Anschlüsse.

Der Widerstand der Anschlussleitungen im 2-Leiter-Modus verursacht einen Fehler der Gebermessung. Die Messgenauigkeit ist daher nicht mehr gewährleistet.

Tabelle A- 155 Rauschminderung und Aktualisierungszeiten für die RTD-Module

Auswahl Unterdrückungsfrequenz	Integrationszeit	Aktualisierungszeit (Sekunden)	
		4-Kanal-Modul	8-Kanal-Modul
400 Hz (2,5 ms)	10 ms <sup>1</sup>	4-/2-Draht: 0.142 3-Draht: 0.285	4-/2-Draht: 0.285 3-Draht: 0.525
60 Hz (16,6 ms)	16,67 ms	4-/2-Draht: 0.222 3-Draht: 0.445	4-/2-Draht: 0.445 3-Draht: 0.845
50 Hz (20 ms)	20 ms	4-/2-Draht: 0.262 3-Draht: .505	4-/2-Draht: 0.524 3-Draht: 1.015
10 Hz (100 ms)	100 ms	4-/2-Draht: 1.222 3-Draht: 2.445	4-/2-Draht: 2.425 3-Draht: 4.845

- <sup>1</sup> Um die Auflösung und Messgenauigkeit des Moduls bei Auswahl des 400-Hz-Filters aufrecht zu erhalten, beträgt die Integrationszeit 10 ms. Mit dieser Auswahl erfolgt auch die Rauschunterdrückung bei 100 Hz und 200 Hz.

ACHTUNG
Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm ggf. berücksichtigt werden. Weil sich die Konfiguration des Moduls auf die Dauer der Initialisierungszeit auswirken kann, prüfen Sie das Verhalten des Moduls in Ihrer Konfiguration. Ggf. können Sie Logik in Ihr Anwenderprogramm aufnehmen, um die Initialisierungszeit des Moduls unterzubringen.

## Darstellung der Analogwerte für RTDs

Die folgenden Tabellen zeigen den digitalisierten Messwert für die Sensoren mit RTD-Standardtemperaturbereich.

Tabelle A- 156 Darstellung von Analogwerten für die Widerstandsthermometer PT 100, 200, 500, 1000 und PT 10, 50, 100, 500 GOST (0,003850) Standard

Pt x00 Standard in °C (1 Ziffer = 0,1 °C)	Einheiten		Pt x00 Standard in °F (1 Ziffer = 0,1 °F)	Einheiten		Bereich
	Dezimal	Hexadezimal		Dezimal	Hexadezimal	
> 1000.0	32767	7FFF	> 1832.0	32767	7FFF	Überlauf
1000.0	10000	2710	1832.0	18320	4790	Oberhalb des Bereichs
:	:	:	:	:	:	
850.1	8501	2135	1562.1	15621	3D05	Bemessungs bereich
850.0	8500	2134	1562.0	15620	3D04	
:	:	:	:	:	:	
-200.0	-2000	F830	-328.0	-3280	F330	

A.9 Digitale Signalboards (SBs)

Pt x00 Standard in °C (1 Ziffer = 0,1 °C)	Einheiten		Pt x00 Standard in °F (1 Ziffer = 0,1 °F)	Einheiten		Bereich
	Dezimal	Hexadezimal		Dezimal	Hexadezimal	
-200.1 :	-2001 :	F82F :	-328.1 :	-3281 :	F32F :	Unterhalb des Bereichs
-243.0	-2430	F682	-405.4	-4054	F02A	
< -243.0	-32768	8000	< -405.4	-32768	8000	Unterlauf

## A.9 Digitale Signalboards (SBs)

### A.9.1 Technische Daten des SB 1221 200 kHz Digitaleingabe

Tabelle A- 157 Allgemeine technische Daten

Technische Daten	SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Bestellnummer	6ES7 221-3BD30-0XB0	6ES7 221-3AD30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21	38 x 62 x 21
Gewicht	35 Gramm	35 Gramm
Leistungsverlust	1,5 W	1,0 W
Stromaufnahme (SM-Bus)	40 mA	40 mA
Stromaufnahme (24 V DC)	7 mA / Eingang + 20 mA	15 mA / Eingang + 15 mA

Tabelle A- 158 Digitaleingänge

Technische Daten	SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Anzahl der Eingänge	4	4
Typ	Quelle	Stromliefernd
Nennspannung	24 V DC bei 7 mA, Nennwert	5 V DC bei 15 mA, Nennwert
Zulässige Dauerspannung	28,8 V DC	6 V DC
Stoßspannung	35 V DC für 0,5 s	6 V
Signal logisch 1 (min.)	L+ minus 10 V DC bei 2,9 mA	L+ minus 2,0 V DC bei 5,1 mA
Signal logisch 0 (max.)	L+ minus 5 V DC bei 1,4 mA	L+ minus 1,0 V DC bei 2,2 mA
HSC Eingangstaktfrequenzen (max.)	Einphasenzähler: 200 kHz A/B-Zähler: 160 kHz	Einphasenzähler: 200 kHz A/B-Zähler: 160 kHz
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1	1
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)



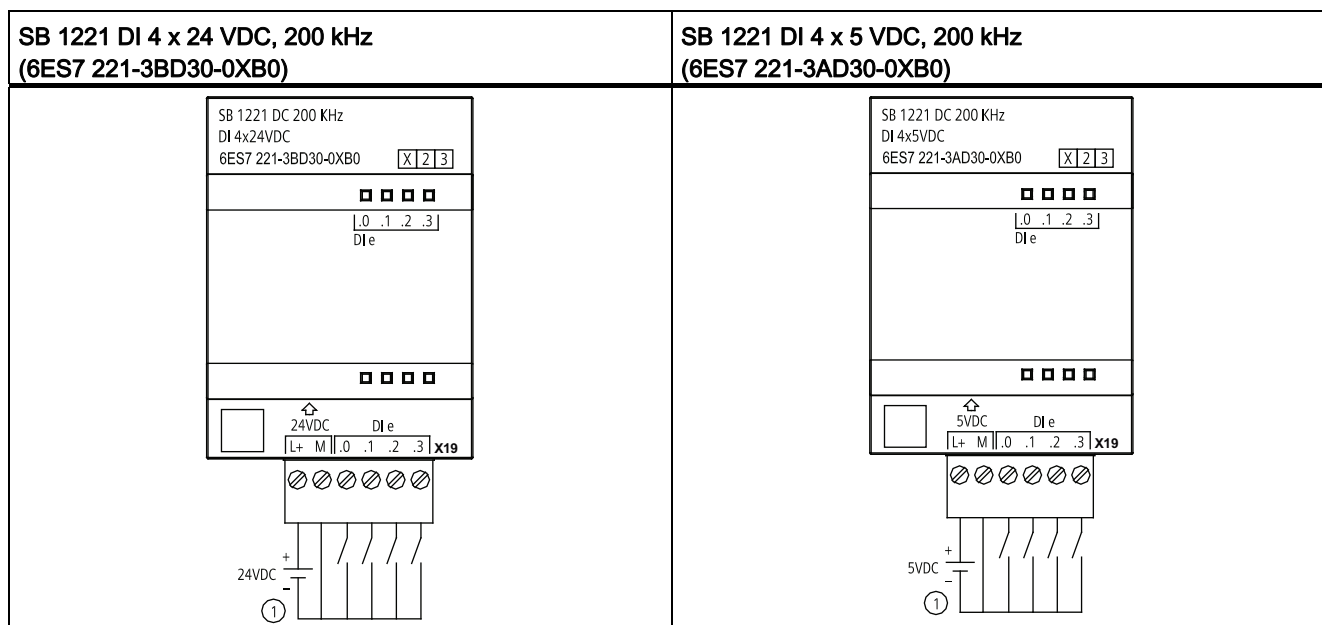
Technische Daten	SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Anzahl gleichzeitig eingeschalteter Eingänge	<ul style="list-style-type: none"> <li>• 2 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 4 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	4
Leitungslänge (Meter)	50 m, geschirmtes, verdrehtes Leiterpaar	50 m, geschirmtes, verdrehtes Leiterpaar

### ACHTUNG

Beim Umschalten von Frequenzen über 20 kHz ist es wichtig, dass die Digitaleingänge Rechtecksignale empfangen. Sie haben folgende Möglichkeiten, die Qualität des Eingangssignals zu verbessern:

- Verkürzen Sie die Leitung auf die Mindestlänge.
- Verwenden Sie statt eines nur stromziehenden Treibers einen stromziehenden/stromliefernden (P-M-schaltenden) Treiber.
- Tauschen Sie das Leitungskabel gegen ein höherwertiges Kabel aus.
- Verringern Sie die Spannung der Schaltkreise/Bauteile von 24 V auf 5 V.
- Fügen Sie am Eingang eine externe Last hinzu.

Tabelle A- 159 Schaltpläne der Digitaleingangs-SBs (200 kHz)



① Unterstützt nur stromliefernde Eingänge

Tabelle A- 160 Anschlussbelegung für das SB 1221 DI 4 x 24 VDC, 200 kHz (6ES7 221-3BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

Tabelle A- 161 Anschlussbelegung für das SB 1221 DI 4 x 5 VDC, 200 kHz (6ES7 221-3AD30-0XB0)

Pin	X19
1	L+ / 5 V DC
2	M / 5 V DC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

## A.9.2 Technische Daten des SB 1222 200 kHz Digitalausgabe

Tabelle A- 162 Allgemeine technische Daten

Technische Daten	SB 1222 DO 4 x 24 V DC, 200 kHz	SB 1222 DO 4 x 5 V DC, 200 kHz
Bestellnummer	6ES7 222-1BD30-0XB0	6ES7 222-1AD30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21	38 x 62 x 21
Gewicht	35 Gramm	35 Gramm
Leistungsverlust	0,5 W	0,5 W
Stromaufnahme (SM-Bus)	35 mA	35 mA
Stromaufnahme (24 V DC)	15 mA	15 mA

Tabelle A- 163 Digitale Ausgänge

Technische Daten	SB 1222 DO 4 x 24 V DC, 200 kHz	SB 1222 DO 4 x 5 V DC, 200 kHz
Ausgänge	4	4
Typ der Ausgänge	MOSFET, elektronisch (stromziehend/stromliefernd) <sup>1</sup>	MOSFET, elektronisch (stromziehend/stromliefernd) <sup>1</sup>
Spannungsbereich	20,4 bis 28,8 V DC	4,25 bis 6,0 V DC
Signal logisch 1 bei max. Strom	L+ minus 1,5 V	L+ minus 0,7 V
Signal logisch 0 bei max. Strom	max. 1,0 V DC	max. 0,2 V DC

<b>Technische Daten</b>	<b>SB 1222 DO 4 x 24 V DC, 200 kHz</b>	<b>SB 1222 DO 4 x 5 V DC, 200 kHz</b>
Strom (max.)	0,1 A	0,1 A
Lampenlast	--	--
Kontaktwiderstand bei EIN	max. 11 $\Omega$	max. 7 $\Omega$
Widerstand bei AUS	max. 6 $\Omega$	max. 0,2 $\Omega$
Kriechstrom pro Ausgang	--	--
Frequenz Impulsgenerator	max. 200 kHz, min. 2 Hz	max. 200 kHz, min. 2 Hz
Einschaltstrom	0,11 A	0,11 A
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1	1
Ströme je Leiter	0,4 A	0,4 A
Induktive Klemmspannung	Keine	Keine
Schaltverzögerung	1,5 $\mu$ s + 300 ns steigend 1,5 $\mu$ s + 300 ns fallend	200 ns + 300 ns steigend 200 ns + 300 ns fallend
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> <li>• 2 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal</li> <li>• 4 bei 55 °C horizontal oder 45 °C vertikal</li> </ul>	4
Leitungslänge (Meter)	50 m, geschirmtes, verdrilltes Leiterpaar	50 m, geschirmtes, verdrilltes Leiterpaar

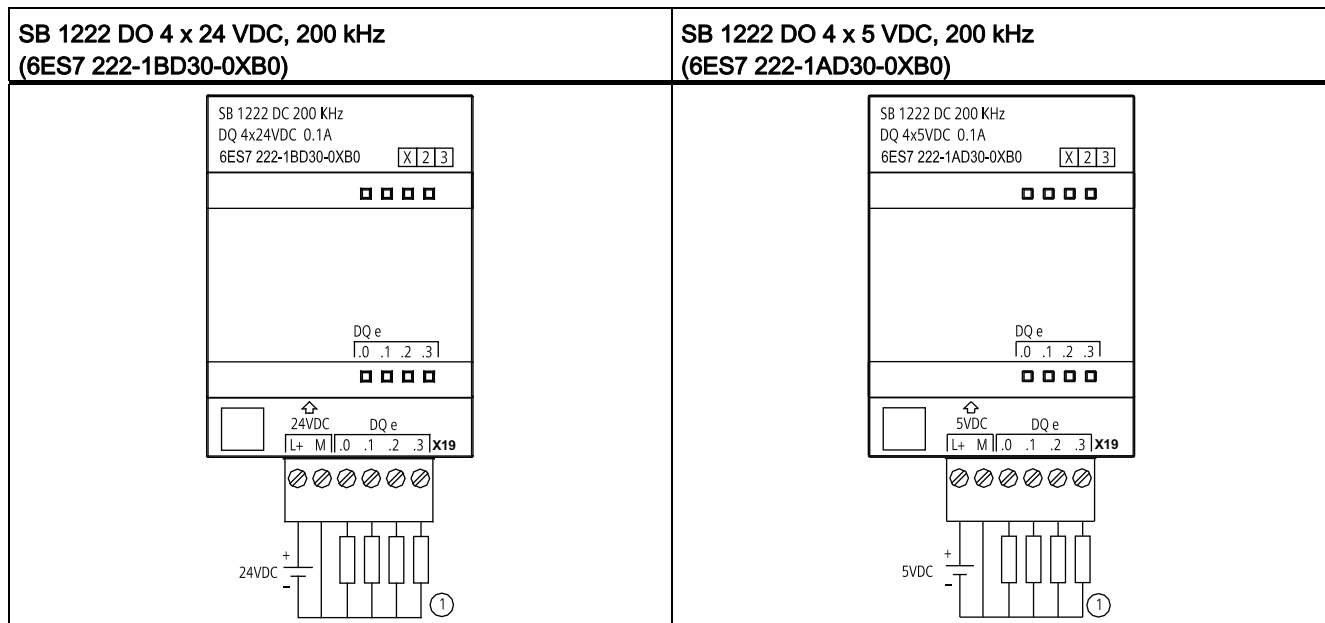
<sup>1</sup> Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

**ACHTUNG**

Beim Umschalten von Frequenzen über 20 kHz ist es wichtig, dass die Digitaleingänge Rechtecksignale empfangen. Sie haben folgende Möglichkeiten, die Qualität des Eingangssignals zu verbessern:

- Verkürzen Sie die Leitung auf die Mindestlänge.
- Verwenden Sie statt eines nur stromziehenden Treibers einen stromziehenden/stromliefernden (P-M-schaltenden) Treiber.
- Tauschen Sie das Leitungskabel gegen ein höherwertiges Kabel aus.
- Verringern Sie die Spannung der Schaltkreise/Bauteile von 24 V auf 5 V.
- Fügen Sie am Eingang eine externe Last hinzu.

Tabelle A- 164 Schaltpläne der Digitalausgangs-SBs (200 kHz)



① Bei stromliefernden Ausgängen "Load" an "-" anschließen (s. Abbildung). Bei stromziehenden Ausgängen "Load" an "+" anschließen. Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

Tabelle A- 165 Anschlussbelegung für das SB 1222 DO 4 x 24 VDC, 200 kHz (6ES7 222-1BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DO e.0
4	DO e.1
5	DO e.2
6	DO e.3

Tabelle A- 166 Anschlussbelegung für das SB 1222 DO 4 x 5 VDC, 200 kHz (6ES7 222-1AD30-0XB0)

Pin	X19
1	L+ / 5 V DC
2	M / 5 V DC
3	DO e.0
4	DO e.1
5	DO e.2
6	DO e.3

### A.9.3 Technische Daten des SB 1223 200 kHz Digitalein-/Digitalausgabe

Tabelle A- 167 Allgemeine technische Daten

Technische Daten	SB 1223 DI 2 x 24 VDC / DO 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DO 2 x 5 VDC, 200 kHz
Bestellnummer	6ES7 223-3BD30-0XB0	6ES7 223-3AD30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21	38 x 62 x 21
Gewicht	35 Gramm	35 Gramm
Leistungsverlust	1,0 W	0,5 W
Stromaufnahme (SM-Bus)	35 mA	35 mA
Stromaufnahme (24 V DC)	7 mA / Eingang + 30 mA	15 mA / Eingang + 15 mA

Tabelle A- 168 Digitaleingänge

Technische Daten	SB 1223 DI 2 x 24 VDC / DO 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DO 2 x 5 VDC, 200 kHz
Anzahl der Eingänge	2	2
Typ	Quelle	Stromliefernd
Nennspannung	24 V DC bei 7 mA, Nennwert	5 V DC bei 15 mA, Nennwert
Zulässige Dauerspannung	28,8 V DC	6 V DC
Stoßspannung	35 V DC für 0,5 s	6 V
Signal logisch 1 (min.)	L+ minus 10 V DC bei 2,9 mA	L+ minus 2,0 V DC bei 5,1 mA
Signal logisch 0 (max.)	L+ minus 5 V DC bei 1,4 mA	L+ minus 1,0 V DC bei 2,2 mA
HSC Eingangstaktfrequenzen (max.)	Einphasenzähler: 200 kHz A/B-Zähler: 160 kHz	Einphasenzähler: 200 kHz A/B-Zähler: 160 kHz
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1 (keine Trennung gegen Ausgang)	1 (keine Trennung gegen Ausgang)
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)

A.9 Digitale Signalboards (SBs)

Technische Daten	SB 1223 DI 2 x 24 VDC / DO 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DO 2 x 5 VDC, 200 kHz
Anzahl gleichzeitig eingeschalteter Eingänge	2	2
Leitungslänge (Meter)	50 m, geschirmtes, verdrilltes Leiterpaar	50 m, geschirmtes, verdrilltes Leiterpaar

Tabelle A- 169 Digitale Ausgänge

Technische Daten	SB 1223 DI 2 x 24 VDC / DO 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DO 2 x 5 VDC, 200 kHz
Ausgänge	2	2
Ausgangstyp	MOSFET, elektronisch (stromziehend/stromliefernd) <sup>1</sup>	MOSFET, elektronisch (stromziehend/stromliefernd) <sup>1</sup>
Spannungsbereich	20,4 bis 28,8 V DC	4,25 bis 6,0 V DC
Nennwert	24 V DC	5 V DC
Signal logisch 1 bei max. Strom	L+ minus 1,5 V	L+ minus 0,7 V
Signal logisch 0 bei max. Strom	max. 1,0 V DC	max. 0,2 V DC
Strom (max.)	0,1 A	0,1 A
Lampenlast	--	--
Kontaktwiderstand bei EIN	max. 11 Ω	max. 7 Ω
Widerstand bei AUS	max. 6 Ω	max. 0,2 Ω
Kriechstrom pro Ausgang	--	--
Frequenz Impulsgenerator	max. 200 kHz, min. 2 Hz	max. 200 kHz, min. 2 Hz
Einschaltstrom	0,11 A	0,11 A
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1 (keine Trennung gegen Eingang)	1 (keine Trennung gegen Eingang)
Ströme je Leiter	0,2 A	0,2 A
Induktive Klemmspannung	Keine	Keine
Schaltverzögerung	1,5 μs + 300 ns steigend 1,5 μs + 300 ns fallend	200 ns + 300 ns steigend 200 ns + 300 ns fallend
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Anzahl gleichzeitig eingeschalteter Ausgänge	2	2
Leitungslänge (Meter)	50 m, geschirmtes, verdrilltes Leiterpaar	50 m, geschirmtes, verdrilltes Leiterpaar

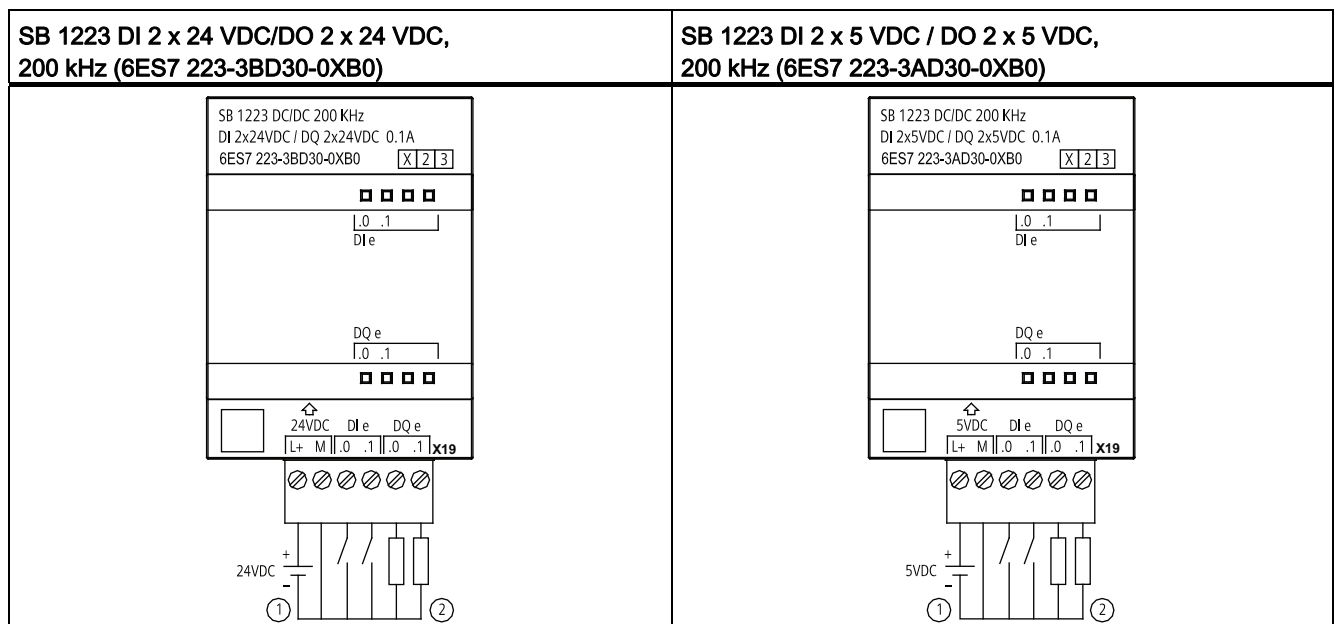
<sup>1</sup> Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

**ACHTUNG**

Beim Umschalten von Frequenzen über 20 kHz ist es wichtig, dass die Digitaleingänge Rechtecksignale empfangen. Sie haben folgende Möglichkeiten, die Qualität des Eingangssignals zu verbessern:

- Verkürzen Sie die Leitung auf die Mindestlänge.
- Verwenden Sie statt eines nur stromziehenden Treibers einen stromziehenden/stromliefernden (P-M-schaltenden) Treiber.
- Tauschen Sie das Leitungskabel gegen ein höherwertiges Kabel aus.
- Verringern Sie die Spannung der Schaltkreise/Bauteile von 24 V auf 5 V.
- Fügen Sie am Eingang eine externe Last hinzu.

Tabelle A- 170 Schaltpläne der digitalen Eingangs-/Ausgangs-SBs (200 kHz)



① Unterstützt nur stromliefernde Eingänge

② Bei stromliefernden Ausgängen "Load" an "-" anschließen (s. Abbildung). Bei stromziehenden Ausgängen "Load" an "+" anschließen. <sup>1</sup> Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

Tabelle A- 171 Anschlussbelegung für das SB 1223 DI 2 x 24 VDC/DO 2 x 24 VDC, 200 kHz (6ES7 223-3BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DI e.0
4	DI e.1
5	DO e.0
6	DO e.1

Tabelle A- 172 Anschlussbelegung für das SB 1223 DI 2 x 5 VDC/DO 2 x 5 VDC, 200 kHz (6ES7 223-3AD30-0XB0)

Pin	X19
1	L+ / 5 V DC
2	M / 5 V DC
3	DI e.0
4	DI e.1
5	DO e.0
6	DO e.1

### A.9.4 Technische Daten SB 1223 2 x 24 V DC Eingang / 2 x 24 V DC Ausgang

Tabelle A- 173 Allgemeine technische Daten

Technische Daten	SB 1223 DI 2 x 24 VDC, DO 2 x 24 V DC
Bestellnummer	6ES7 223-0BD30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	40 Gramm
Leistungsverlust	1,0 W
Stromaufnahme (SM-Bus)	50 mA
Stromaufnahme (24 V DC)	4 mA / Eingang

Tabelle A- 174 Digitaleingänge

Technische Daten	SB 1223 DI 2 x 24 VDC, DO 2 x 24 V DC
Anzahl der Eingänge	2
Typ	IEC Typ 1 stromziehend
Nennspannung	24 V DC bei 4 mA, Nennwert
Zulässige Dauerspannung	max. 30 V DC



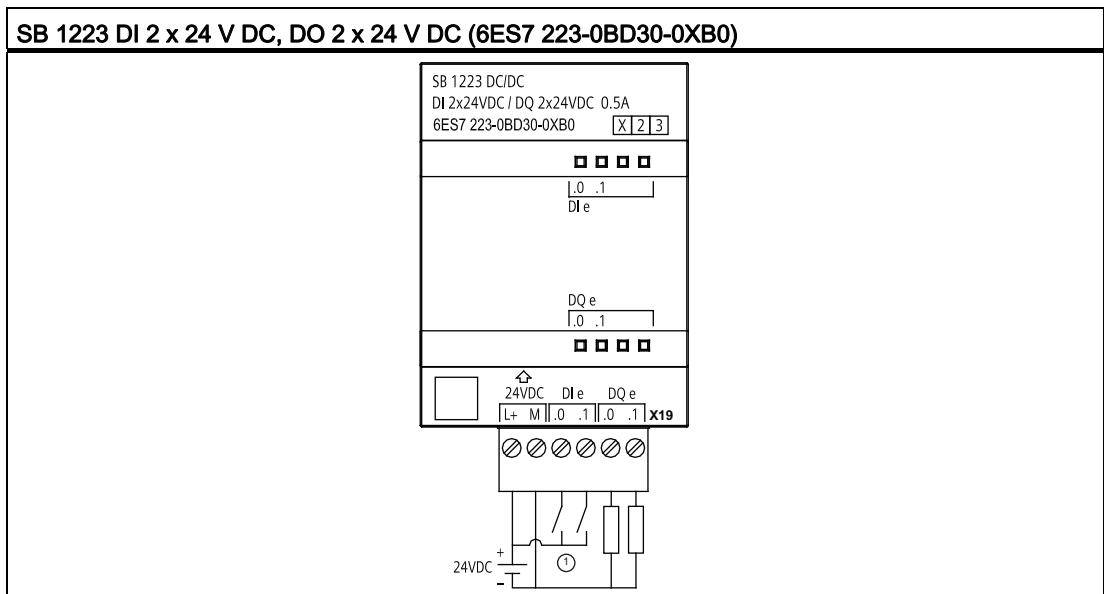
Technische Daten	SB 1223 DI 2 x 24 VDC, DO 2 x 24 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
HSC Eingangstaktfrequenzen (max.)	20 kHz (15 bis 30 V DC) 30 kHz (15 bis 26 V DC)
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 2
Anzahl gleichzeitig eingeschalteter Eingänge	2
Leitungslänge (Meter)	500 geschirmt, 300 ungeschirmt

Tabelle A- 175 Digitale Ausgänge

Technische Daten	SB 1223 DI 2 x 24 VDC, DO 2 x 24 V DC
Ausgänge	2
Ausgangstyp	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	max. 0,1 V DC
Strom (max.)	0,5 A
Lampenlast	5 W
Kontaktwiderstand bei EIN	max. 0,6 Ω
Kriechstrom pro Ausgang	max. 10 µA
Frequenz Impulsfolgeausgang (PTO)	max. 20 kHz, min. 2 Hz <sup>1</sup>
Einschaltstrom	5 A für max. 100 ms
Überlastschutz	Nein
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1
Ströme je Leiter	1 A
Induktive Klemmspannung	L+ minus 48 V, 1 W Verlustleistung
Schaltverzögerung	max. 2 µs von Aus nach Ein max. 10 µs von Ein nach Aus
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Anzahl gleichzeitig eingeschalteter Ausgänge	2
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt

<sup>1</sup> Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

Tabelle A- 176 Schaltplan des digitalen Eingangs-/Ausgangs-SBs



① Unterstützt nur stromziehende Eingänge

Tabelle A- 177 Anschlussbelegung für das SB 1223 DI 2 x 24 VDC/DO 2 x 24 VDC (6ES7 223-0BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DI e.0
4	DI e.1
5	DO e.0
6	DO e.1

## A.10 Analoge Signalboards (SBs)

### A.10.1 Technische Daten des SB 1231 1 Analogeingang

**Hinweis**

Um diesen SB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A- 178 Allgemeine technische Daten

<b>Technische Daten</b>	<b>SB 1231 AI 1 x 12 Bit</b>
Bestellnummer	6ES7 231-4HA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	35 Gramm
Leistungsverlust	0,4 W
Stromaufnahme (SM-Bus)	55 mA
Stromaufnahme (24 V DC)	Keine

Tabelle A- 179 Analoge Eingänge

<b>Technische Daten</b>	<b>SB 1231 AI 1x12 Bit</b>
Anzahl der Eingänge	1
Typ	Spannung oder Strom (differential)
Bereich	$\pm 10$ V, $\pm 5$ V, $\pm 2,5$ oder 0 bis 20 mA
Auflösung	11 Bit + Vorzeichenbit
Vollausschlag (Datenwort)	-27.648 bis 27.648
Überbereich/Unterbereich (Datenwort)	Spannung: 32.511 bis 27.649 / -27.649 bis -32.512 Strom: 32.511 bis 27.649/0 bis -4.864 (siehe Darstellung Analogeingang für Spannung und Darstellung Analogeingang für Strom).
Überlauf/Unterlauf (Datenwort)	Spannung: 32.767 bis 32.512/-32.513 bis -32.768 Strom: 32.767 bis 32.512/-4.865 bis -32.768 (siehe Darstellung Analogeingang für Spannung und Darstellung Analogeingang für Strom).
Maximale Stehspannung/-strom	$\pm 35$ V/ $\pm 40$ mA
Glättung	Keine, schwach, mittel oder stark (siehe Antwortzeiten Analogeingang für Schrittantwortzeit).
Rauschunterdrückung	400, 60, 50 oder 10 Hz (siehe Ansprechzeiten Analogeingang für Abtastraten).
Genauigkeit (25 °C / -20 bis 60 °C)	$\pm 0,3\%$ / $\pm 0,6\%$ des Vollausschlags
Eingangsimpedanz	
Differential	Spannung: 220 k $\Omega$ ; Strom: 250 $\Omega$
Gleichtakt	Spannung: 55 k $\Omega$ ; Strom: 55 k $\Omega$
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Messprinzip	Istwertumwandlung
Gleichtaktunterdrückung	400 dB, Nennwert bei 60 Hz
Betriebssignalebereich	Signal- plus Gleichtaktspannung muss kleiner als +35 V und größer als -35 V sein
Elektrische Trennung (Feld zu Logik)	Keine
Leitungslänge (Meter)	100 m, verdrillt und geschirmt

Tabelle A- 180 Diagnose

Technische Daten	SB 1231 AI 1 x 12 Bit
Überlauf/Unterlauf	Ja
24 V DC Niederspannung	Nein

Tabelle A- 181 Schaltplan des Analogeingangs-SBs

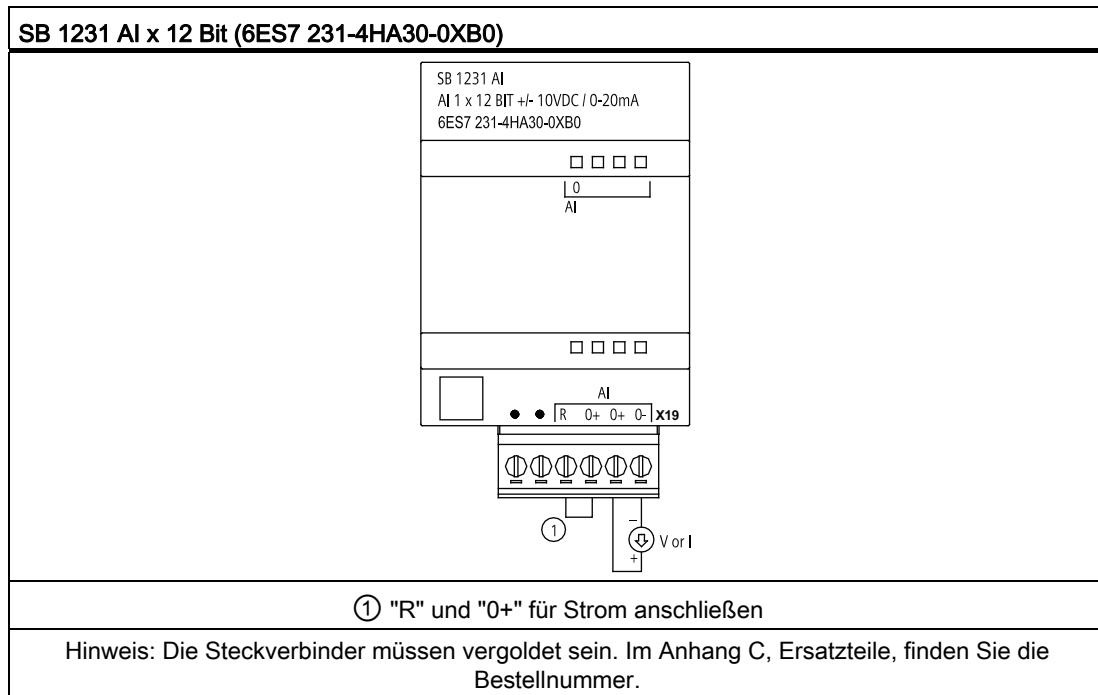


Tabelle A- 182 Anschlussbelegung für das SB 1231 AI x 12 Bit (6ES7 231-4HA30-0XB0)

Pin	X19 (vergoldet)
1	Kein Anschluss
2	Kein Anschluss
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-

## A.10.2 Technische Daten des SB 1232 1 Analogausgabe

Tabelle A- 183 Allgemeine technische Daten

Technische Daten	SB 1232 AO 1 x 12 Bit
Bestellnummer	6ES7 232-4HA30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21
Gewicht	40 Gramm
Leistungsverlust	1,5 W
Stromaufnahme (SM-Bus)	15 mA
Stromaufnahme (24 V DC)	40 mA (ohne Last)

Tabelle A- 184 Analoge Ausgänge

Technische Daten	SB 1232 AO 1 x 12 Bit
Ausgänge	1
Typ	Spannung oder Strom
Bereich	$\pm 10$ V oder 0 bis 20 mA
Auflösung	Spannung: 12 Bits Strom: 11 Bits
Vollausschlag (Datenwort)	Spannung: -27.648 bis 27.648
Siehe Abschnitt der Ausgangsbereiche für Spannung und Strom (Seite 849).	Strom: 0 bis 27.648
Genauigkeit (25 °C / -20 bis 60 °C)	$\pm 0,5\%$ / $\pm 1\%$ des Vollausschlags
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 $\mu$ S (R), 750 $\mu$ S (1 $\mu$ F) Strom: 600 $\mu$ S (1 mH), 2 ms (10 mH)
Lastimpedanz	Spannung: $\geq 1000 \Omega$ Strom: $\leq 600 \Omega$
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Elektrische Trennung (Feld zu Logik)	Keine
Leitungslänge (Meter)	100 m, verdreht und geschirmt

Tabelle A- 185 Diagnose

Technische Daten	SB 1232 AO 1 x 12 Bit
Überlauf/Unterlauf	Ja
Erdschluss (nur Spannungsmodus)	Ja
Drahtbruch (nur Strommodus)	Ja

Tabelle A- 186 Schaltplan für das SB 1232 AO 1 x 12 Bit

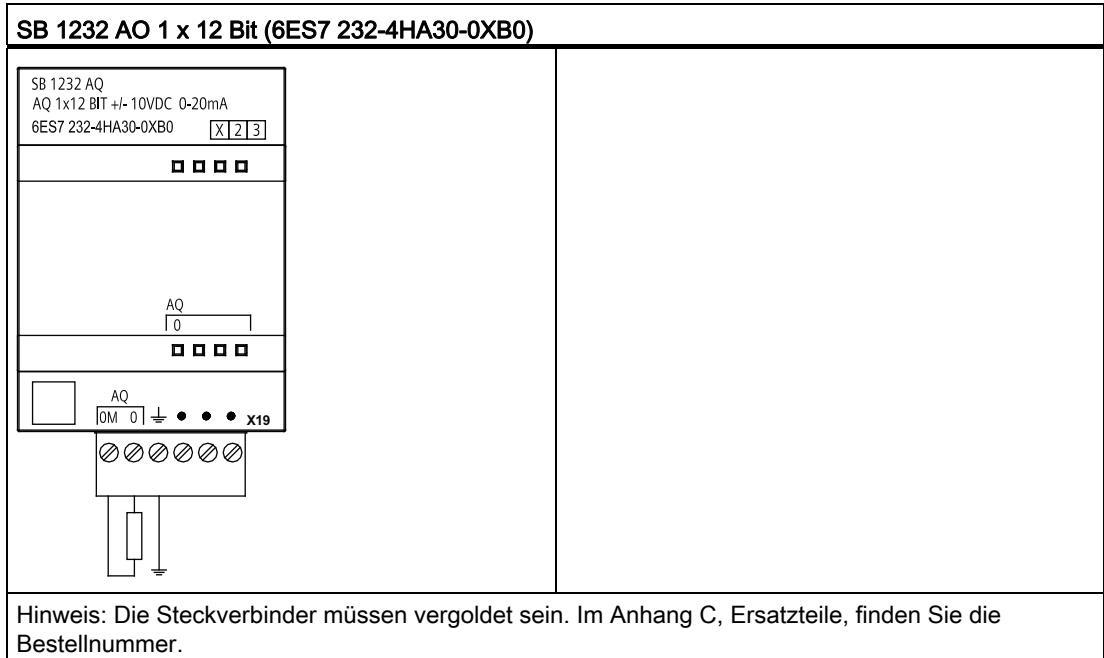


Tabelle A- 187 Anschlussbelegung für das SB 1232 AO 1 x 12 Bit (6ES7 232-4HA30-0XB0)

Pin	X19 (vergoldet)
1	AO 0M
2	AO 0
3	GND
4	Kein Anschluss
5	Kein Anschluss
6	Kein Anschluss

## A.10.3 Messbereiche der analogen Eingänge und Ausgänge

### A.10.3.1 Schrittantwort der analogen Eingänge

Tabelle A- 188 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Auswahl Integrationszeit			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Keine (1 Zyklus): Keine Mittelwertbildung	4,5 ms	18,7 ms	22,0 ms	102 ms
Schwach (4 Zyklen): 4 Abtastwerte	10,6 ms	59,3 ms	70,8 ms	346 ms
Mittel (16 Zyklen): 16 Abtastwerte	33,0 ms	208 ms	250 ms	1240 ms
Stark (32 Zyklen): 32 Abtastwerte	63,0 ms	408 ms	490 ms	2440 ms
<b>Abtastzeit</b>	<b>0,156 ms</b>	<b>1,042 ms</b>	<b>1,250 ms</b>	<b>6,250 ms</b>

### A.10.3.2 Abtastzeit und Aktualisierungszeiten der Analogeingänge

Tabelle A- 189 Abtastzeit und Aktualisierungszeit

Auswahl	Abtastzeit	Aktualisierungszeit SB
400 Hz (2,5 ms)	0,156 ms	0,156 ms
60 Hz (16,6 ms)	1,042 ms	1,042 ms
50 Hz (20 ms)	1,250 ms	1,25 ms
10 Hz (100 ms)	6,250 ms	6,25 ms

### A.10.3.3 Messbereiche der analogen Eingänge für Spannung

Tabelle A- 190 Darstellung Analogeingang für Spannung

System		Messbereich Spannung							
Dezimal	Hexadezimal	±10 V	±5 V	±2,5 V	±1,25 V		0 bis 10 V		
32767	7FFF	11,851 V	5,926 V	2,963 V	1,481 V	Überlauf	11,851 V	Überlauf	
32512	7F00								
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Überschwingbereich	11,759 V	Überschwingbereich	
27649	6C01								
27648	6C00	10 V	5 V	2,5 V	1,250 V	Bemessungsbereich	10 V	Bemessungsbereich	
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V		7,5 V		
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV		361,7 µV		
0	0	0 V	0 V	0 V	0 V		0 V		
-1	FFFF						Negative Werte werden nicht unterstützt		
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V				
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V				
-27649	93FF								
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	Unterschwingbereich			
-32513	80FF								
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	Unterlauf			

### A.10.3.4 Messbereiche der analogen Eingänge für Strom

Tabelle A- 191 Darstellung Analogeingang für Strom

System		Messbereich Strom		
Dezimal	Hexidezimal	0 mA bis 20 mA	4 mA bis 20 mA	
32767	7FFF	23,70 mA	22,96 mA	Überlauf
32512	7F00			
32511	7EFF	23,52 mA	22,81 mA	Überschwingbereich
27649	6C01			
27648	6C00	20 mA	20 mA	Nennbereich
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			
-4864	ED00	-3,52 mA	1,185 mA	Unterschwingbereich
-4865	ECFF			
-32768	8000			Unterlauf



### A.10.3.5 Ausgangsmessbereiche (AO) für Spannung und Strom (SB und SM)

Tabelle A- 192 Darstellung Analogausgang für Spannung

System		Spannungsausgangsbereich	
Dezimal	Hexadezimal	±10 V	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	
32511	7EFF	11,76 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11,76 V	Unterschwingbereich
-32513	80FF	Siehe Hinweis 1	
-32768	8000	Siehe Hinweis 1	

<sup>1</sup> Bei Überlauf oder Unterlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration des Analogsignalmoduls. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: Ersatzwert aufschalten oder Letzten Wert halten.

Tabelle A- 193 Darstellung Analogausgang für Strom

System		Stromausgangsbereich	
Dezimal	Hexadezimal	0 mA bis 20 mA	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	
32511	7EFF	23,52 mA	Überschwingbereich
27649	6C01		
27648	6C00	20 mA	Bemessungsbereich
20736	5100	15 mA	
1	1	723,4 nA	
0	0	0 mA	

<sup>1</sup> Bei Überlauf oder Unterlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration des Analogsignalmoduls. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: Ersatzwert aufschalten oder Letzten Wert halten.

### A.10.4 Thermoelement-SBs

#### A.10.4.1 Technische Daten des SB 1231 1 Analogeingang Thermoelement

**Hinweis**

Um diesen SB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A- 194 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 x 16 Bit Thermoelement
Bestellnummer	6ES7 231-5QA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	35 Gramm
Leistungsverlust	0,5 W
Stromaufnahme (SM-Bus)	5 mA
Stromaufnahme (24 V DC)	20 mA

Tabelle A- 195 Analoge Eingänge

Technische Daten	SB 1231 AI 1x16 Bit Thermoelement				
Anzahl der Eingänge	1				
Typ	Potentialfrei, TC und mV				
Bereich	Siehe Filterauswahltabelle Thermoelement (Seite 851). <ul style="list-style-type: none"> <li>• Nennbereich (Datenwort)</li> <li>• Überlauf/Unterlauf (Datenwort)</li> <li>• Überlauf/Unterlauf (Datenwort)</li> </ul>				
Auflösung	<table border="0"> <tr> <td style="border-right: 1px solid black;">Temperaturbereich</td> <td>0,1 °C/0,1 °F</td> </tr> <tr> <td style="border-right: 1px solid black;">Spannung</td> <td>15 Bit plus Vorzeichen</td> </tr> </table>	Temperaturbereich	0,1 °C/0,1 °F	Spannung	15 Bit plus Vorzeichen
Temperaturbereich	0,1 °C/0,1 °F				
Spannung	15 Bit plus Vorzeichen				
Max. Stehspannung	±35 V				
Rauschunterdrückung	85 dB für die gewählte Filtereinstellung (10 Hz, 50 Hz, 60 Hz, 400 Hz)				
Gleichtaktunterdrückung	> 120 dB bei 120 V AC				
Impedanz	≥ 10 MΩ				
Genauigkeit	Siehe Thermoelement-Auswahltabelle (Seite 851).				
Wiederholgenauigkeit	±0,05 % Vollausschlag				
Messprinzip	Integrierend				
Aktualisierungszeit Modul	Siehe Filterauswahltabelle Thermoelement (Seite 851).				
Fehler kalte Verbindungsstelle	±1,5 °C				
Elektrische Trennung (Feld zu Logik)	500 V AC				

Technische Daten	SB 1231 AI 1x16 Bit Thermoelement
Leitungslänge (Meter)	Max. 100 m zum Geber
Leitungswiderstand	max. 100 Ω

Tabelle A- 196 Diagnose

Technische Daten	SB 1231 AI 1 x 16 Bit Thermoelement
Überlauf/Unterlauf <sup>1</sup>	Ja
Drahtbruch <sup>2</sup>	Ja

<sup>1</sup> Die Informationen der Diagnosealarme "Überlauf" und "Unterlauf" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.

<sup>2</sup> Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Das analoge Thermoelement-Signalmodul SM 1231 TC misst den Wert der an die Moduleingänge angeschlossenen Spannung.

Das analoge Thermoelement-Signalboard SB 1231 misst den Wert der an die Signalboardeingänge angeschlossenen Spannung. Als Temperaturmessart sind entweder "Thermoelement" oder "Spannung" möglich.

- "Thermoelement": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).
- "Spannung": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.

#### A.10.4.2 Grundlegende Funktionsweise eines Thermoelements

Thermoelemente entstehen, wenn zwei unterschiedliche Metalle elektrisch miteinander verbunden werden. Dadurch wird eine Spannung erzeugt, die proportional zu der Temperatur der Verbindungsstelle ist. Es handelt sich um eine geringe Spannung. Ein Mikrovolt kann viele Grade darstellen. Grundlage für die Temperaturmessung mit Thermoelementen sind das Messen der Spannung eines Thermoelements, das Kompensieren von zusätzlichen Verbindungsstellen und das Linearisieren der Ergebnisse.

Wenn Sie ein Thermoelement an das SM 1231 Thermoelementmodul anschließen, werden die beiden Leitungen der unterschiedlichen Metalle am Signalanschluss des Moduls angeschlossen. Die Stelle, an der die beiden unterschiedlichen Leitungen miteinander verbunden werden, bildet den Sensor des Thermoelements.

Zwei weitere Thermoelemente entstehen an der Stelle, an der die unterschiedlichen Leitungen an den Signalanschluss angeschlossen werden. Die Temperatur des Anschlusses erzeugt eine Spannung, die zu der Spannung des Thermoelementsensors addiert wird. Wird diese Spannung nicht ausgeglichen, weicht die ausgegebene Temperatur von der Temperatur des Sensors ab.

A.10 Analoge Signalboards (SBs)

Durch die Kompensation der kalten Verbindungsstelle wird das Thermoelement am Anschluss ausgeglichen. Tabellen für Thermoelemente basieren auf einer Bezugstemperatur an der Verbindungsstelle, üblicherweise Null Grad Celsius. Durch die Kompensation der kalten Verbindungsstelle wird der Anschluss auf Null Grad Celsius kompensiert. Die Spannung, die durch das Thermoelement des Anschlusses addiert wird, wird durch die Kompensation der kalten Verbindungsstelle korrigiert. Die Temperatur des Moduls wird innen gemessen und dann in einen Wert umgewandelt, der zur Sensorumsetzung addiert wird. Die korrigierte Sensorumsetzung wird dann mittels der Thermoelement-Tabellen linearisiert.

Für die optimale Funktionsweise der Kompensation der kalten Verbindungsstelle muss sich das Thermoelementmodul in einer thermisch stabilen Umgebung befinden. Ein langsame Veränderung (weniger als 0,1 °C/Minute) der Temperatur in der Umgebung des Moduls wird innerhalb der Modulspezifikation korrekt ausgeglichen. Auch Luftbewegungen am Modul verursachen Fehler bei der Kompensation der kalten Verbindungsstelle.

Ist eine bessere Kompensation von Fehlern an der kalten Verbindungsstelle erforderlich, kann eine externe isothermische Klemmenleiste verwendet werden. Mit dem Thermoelementmodul kann eine auf 0 °C bezogene oder eine auf 50 °C bezogene Klemmenleiste eingesetzt werden.

**Auswahltabelle für das SB 1231 Thermoelement**

Die Bereiche und Genauigkeit der verschiedenen vom SB 1231 Thermoelement-Signalboard unterstützten Thermoelementtypen entnehmen Sie bitte der unten stehenden Tabelle.

Tabelle A- 197 SB 1231 Thermoelement-Auswahltabelle

Thermoelementtyp	Minimum unterer Bereich <sup>1</sup>	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich <sup>2</sup>	Normalbereich <sup>3</sup> Genauigkeit bei 25 °C	Normalbereich <sup>3</sup> Genauigkeit -20 °C bis 60 °C
J	-210,0 °C	-150,0 °C	1200,0 °C	1450,0 °C	±0,3 °C	±0,6°C
K	-270,0 °C	-200,0 °C	1372,0 °C	1622,0 °C	±0,4 °C	±1,0 °C
T	-270,0 °C	-200,0 °C	400,0 °C	540,0 °C	±0,5 °C	±1,0 °C
E	-270,0 °C	-200,0 °C	1000,0 °C	1200,0 °C	±0,3°C	±0,6 °C
R & S	-50,0 °C	100,0 °C	1768,0 °C	2019,0 °C	±1,0 °C	±2,5 °C
N	-270,0 °C	0,0 °C	1300,0 °C	1550,0 °C	±1,0 °C	±1,6 °C
C	0,0 °C	100,0 °C	2315,0 °C	2500,0 °C	±0,7 °C	±2,7°C
TXK/XK(L)	-200,0 °C	-150,0 °C	800,0 °C	1050,0 °C	±0,6°C	±1,2 °C
Spannung	-32511	-27648 -80 mV	27648 80 mV	32511	±0.05%	±0.1%

- <sup>1</sup> Die Thermoelementwerte unterhalb des Minimums für den unteren Bereich werden als -32768 ausgegeben.
- <sup>2</sup> Die Thermoelementwerte oberhalb des Minimums für den oberen Bereich werden als 32767 ausgegeben.
- <sup>3</sup> Der interne Fehler an der kalten Verbindungsstelle beträgt ±1,5 °C für alle Bereiche. Dieser Wert ist zum in dieser Tabelle aufgeführten Fehler zu addieren. Das Signalboard benötigt eine Aufwärmzeit von mindestens 30 Minuten, bis die hier genannten Spezifikationen erfüllt werden.

Tabelle A- 198 Filterauswahltabelle für das SB 1231 Thermoelement

Unterdrückungsfrequenz (Hz)	Integrationszeit (ms)	Aktualisierungszeit Signalboard (Sekunden)
10	100	0.306
50	20	0.066
60	16.67	0.056
400 <sup>1</sup>	10	0.036

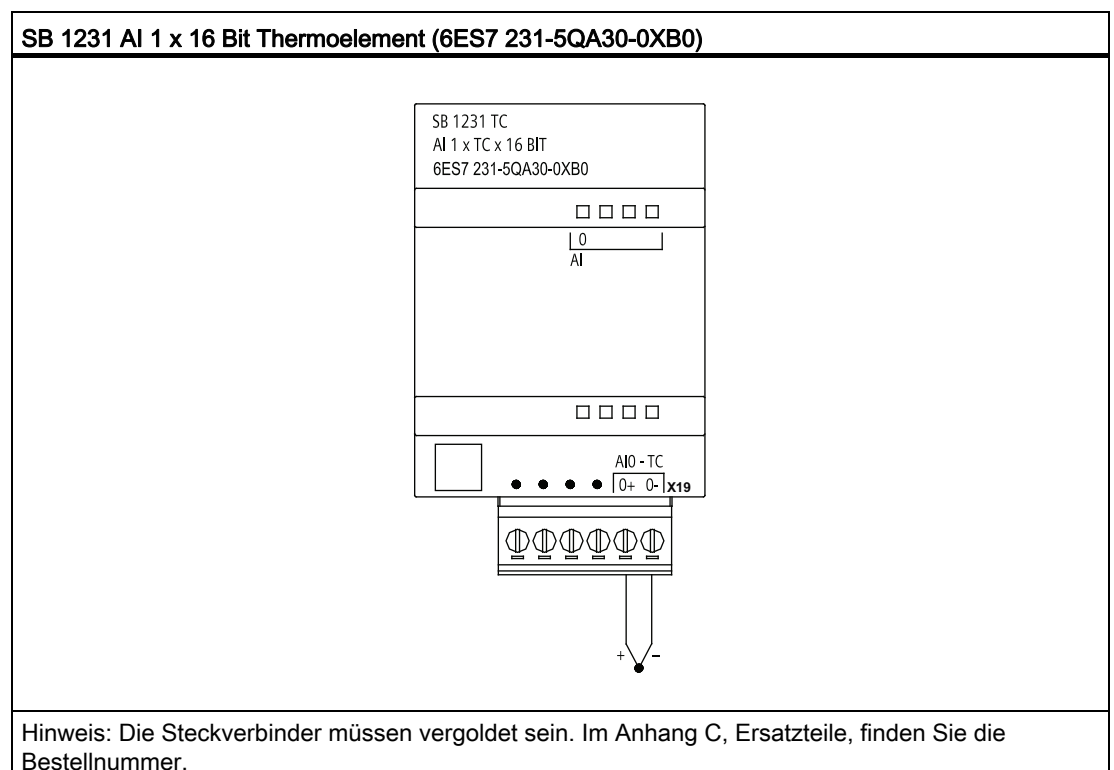
<sup>1</sup> Um die Auflösung und Messgenauigkeit des Moduls bei Auswahl der 400-Hz-Unterdrückung aufrecht zu erhalten, beträgt die Integrationszeit 10 ms. Mit dieser Auswahl erfolgt auch die Rauschunterdrückung bei 100 Hz und 200 Hz.

Für die Messung von Thermoelementen wird eine Integrationszeit von 100 ms empfohlen. Niedriger eingestellte Integrationszeiten führen zu einem höheren Wiederholgenauigkeitsfehler der Temperaturmessungen.

#### Hinweis

Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm ggf. berücksichtigt werden.

Tabelle A- 199 Schaltplan des Analogeingangs-SBs Thermoelement



A.10 Analoge Signalboards (SBs)

Tabelle A- 200 Anschlussbelegungen für das SB 1231 AI 1 x 16 Bit Thermoelement (6ES7 231-5QA30-0XB0)

Pin	X19 (vergoldet)
1	Kein Anschluss
2	Kein Anschluss
3	Kein Anschluss
4	Kein Anschluss
5	AI 0- /TC
6	AI 0+ /TC

A.10.5 RTD-SBs

A.10.5.1 Technische Daten des SB 1231 1 Analogeingang RTD

**Hinweis**

Um diesen SB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A- 201 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 x 16 Bit RTD
Bestellnummer	6ES7 231-5PA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 2
Gewicht	35 Gramm
Leistungsverlust	0,7 W
Stromaufnahme (SM-Bus)	5 mA
Stromaufnahme (24 V DC)	25 mA

Tabelle A- 202 Analoge Eingänge

Technische Daten	SB 1231 AI 1 x 16 Bit RTD
Anzahl der Eingänge	1
Typ	Modulreferenz RTD und Ohm
Bereich	Siehe Auswahltabellen (Seite 857).
Auflösung	Nennbereich (Datenwort)
	Überlauf/Unterlauf (Datenwort)
	Überlauf/Unterlauf (Datenwort)
Temperatur	0,1 °C/0,1 °F
	Spannung

Technische Daten	SB 1231 AI 1 x 16 Bit RTD
Max. Stehspannung	±35 V
Rauschunterdrückung	85 dB (10 Hz, 50 Hz, 60 Hz, 400 Hz)
Gleichtaktunterdrückung	> 120 dB
Impedanz	≥ 10 MΩ
Genauigkeit	Siehe Auswahltabellen (Seite 857).
Wiederholgenauigkeit	±0,05 % Vollausschlag
Maximale Verlustleistung Geber	0.5 m W
Messprinzip	Integrierend
Aktualisierungszeit Modul	Siehe Auswahltable (Seite 857).
Elektrische Trennung (Feld zu Logik)	500 V AC
Leitungslänge (Meter)	Max. 100 m zum Geber
Leitungswiderstand	20 Ω, 2,7 für max. 10 Ω RTD

Tabelle A- 203 Diagnose

Technische Daten	SB 1231 AI 1 x 16 Bit RTD
Überlauf/Unterlauf <sup>1, 2</sup>	Ja
Drahtbruch <sup>3</sup>	Ja

<sup>1</sup> Die Informationen der Diagnosealarme "Überlauf" und "Unterlauf" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.

<sup>2</sup> Für Widerstandsmessbereiche ist die Prüfung auf Unterlauf grundsätzlich nicht aktiviert.

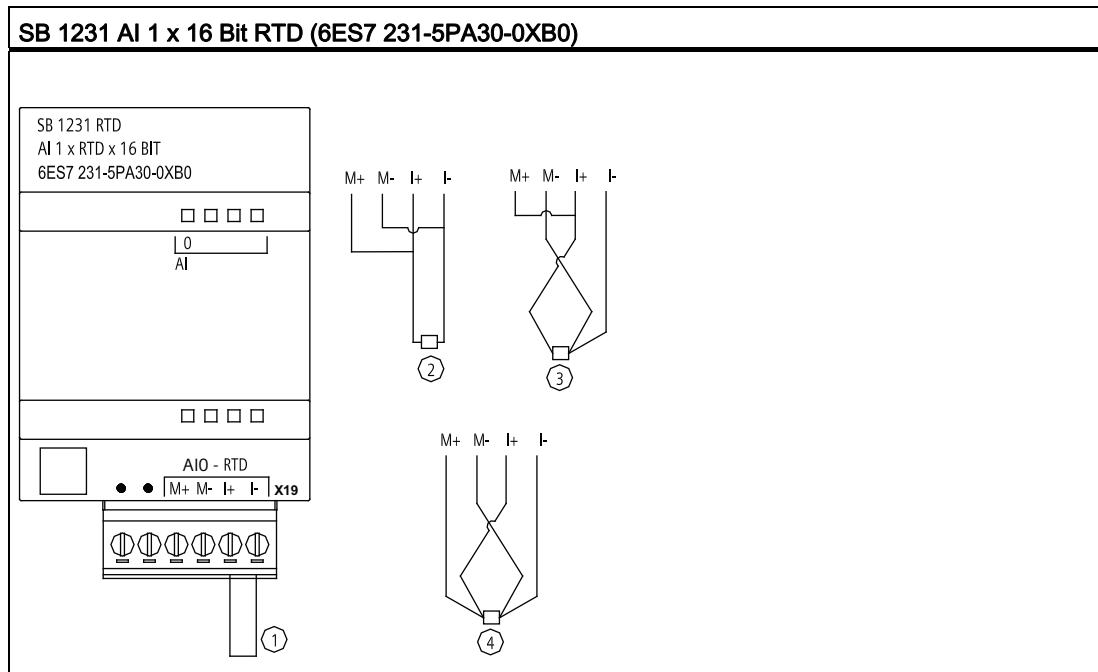
<sup>3</sup> Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Das analoge Thermoelement-Signalboard SM 1231 RTD misst den Wert des an die Signalboardeingänge angeschlossenen Widerstands. Als Messart kann entweder "Widerstand" oder "Thermischer Widerstand" ausgewählt werden.

- "Widerstand": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.
- "Thermischer Widerstand": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).

Das Signalboard SB 1231 RTD unterstützt Messungen über 2-Leiter-, 3-Leiter- und 4-Leiter-Anschlüsse zum Geberwiderstand.

Tabelle A- 204 Schaltplan des SB 1231 AI 1 x 16 Bit RTD



- ① Nicht belegten RTD-Eingang zurückschleifen
- ① 2Leiter-RTD
- ① 3Leiter-RTD
- ① 4Leiter-RTD

Hinweis: Die Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Bestellnummer.

Tabelle A- 205 Anschlussbelegungen für das SB 1231 AI 1 x 16 Bit RTD (6ES7 231-5PA30-0XB0)

Pin	X19 (vergoldet)
1	Kein Anschluss
2	Kein Anschluss
3	AI 0 M+ /RTD
4	AI 0 M- /RTD
5	AI 0 I+ /RTD
6	AI 0 I- /RTD



### A.10.5.2 Auswahltabellen für das SB 1231 RTD

Tabelle A- 206 Bereiche und Genauigkeit für die verschiedenen Geber, die von den RTD-Modulen unterstützt werden

Temperaturkoeffizient	RTD-Typ	Minimum unterer Bereich <sup>1</sup>	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich <sup>2</sup>	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 10	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±1,0 °C	±2,0 °C
	Pt 50	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 100						
	Pt 200						
	Pt 500						
Pt 0,003902 Pt 0,003916 Pt 0,003920	Pt 100	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±1,0 °C	±2,0 °C
	Pt 50	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±0,8 °C	±1,6 °C
	Pt 100						
	Pt 500						
Ni 0,006720 Ni 0,006180	Ni 100	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0,005000	LG-Ni 1000	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
Ni 0,006170	Ni 100	-105,0 °C	-60,0 °C	180,0 °C	212,4 °C	±0,5 °C	±1,0 °C
Cu 0,004270	Cu 10	-240,0 °C	-200,0 °C	260,0 °C	312,0 °C	±1,0 °C	±2,0 °C
Cu 0,004260	Cu 10	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±0,6 °C	±1,2 °C
	Cu 100						
Cu 0,004280	Cu 10	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±0,7 °C	±1,4 °C
	Cu 100						

<sup>1</sup> Die RTD-Werte unterhalb des Minimums für den unteren Bereich werden als -32768 ausgegeben.

<sup>2</sup> Die RTD-Werte oberhalb des Maximums für den oberen Bereich werden als +32768 ausgegeben.

Tabelle A- 207 Beständigkeit gegen:

Bereich	Minimum unterer Bereich	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich <sup>1</sup>	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
150 Ω	nicht zutreffend	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	±0.05%	±0.1%
300 Ω	nicht zutreffend	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	±0.05%	±0.1%
600 Ω	nicht zutreffend	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	±0.05%	±0.1%

<sup>1</sup> Die Widerstandswerte oberhalb des Maximums für den oberen Bereich werden als 32767 ausgegeben.

**Hinweis**

Für alle aktivierte Kanäle ohne angeschlossenen Geber meldet das Modul 32767. Wenn außerdem die Prüfung auf offene Leitungen aktiviert ist, blinken am Modul die entsprechenden roten LED.

Wenn RTD-Bereiche von 500 Ω und 1000 Ω mit anderen Widerständen niederen Werts verwendet werden, kann sich der Fehler auf den zweifachen spezifizierten Fehler erhöhen.

Optimale Genauigkeit für die 10 Ω-RTD-Bereiche ermöglichen 4-Leiter-Anschlüsse.

Der Widerstand der Anschlussleitungen im 2-Leiter-Modus verursacht einen Fehler der Gebermessung. Die Messgenauigkeit ist daher nicht mehr gewährleistet.

Tabelle A- 208 Rauschminderung und Aktualisierungszeiten für die RTD-Module

Auswahl Unterdrückungsfrequenz z	Integrationszeit	1-kanaliges 4/2-Draht-Modul Aktualisierungszeit (Sekunden)	1-kanaliges 3-Draht-Modul Aktualisierungszeit (Sekunden)
400 Hz (2,5 ms)	10 ms <sup>1</sup>	0.036	0.071
60 Hz (16,6 ms)	16,67 ms	0.056	0.111
50 Hz (20 ms)	20 ms	0.066	1.086
10 Hz (100 ms)	100 ms	0.306	0.611

<sup>1</sup> Um die Auflösung und Messgenauigkeit des Moduls bei Auswahl des 400-Hz-Filters aufrecht zu erhalten, beträgt die Integrationszeit 10 ms. Mit dieser Auswahl erfolgt auch die Rauschunterdrückung bei 100 Hz und 200 Hz.

**ACHTUNG**

Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm ggf. berücksichtigt werden. Weil sich die Konfiguration des Moduls auf die Dauer der Initialisierungszeit auswirken kann, prüfen Sie das Verhalten des Moduls in Ihrer Konfiguration. Ggf. können Sie Logik in Ihr Anwenderprogramm aufnehmen, um die Initialisierungszeit des Moduls unterzubringen.

## A.11 BB 1297 Batterieboard

### BB 1297 Batterieboard

Das S7-1200 BB 1297 Batterieboard dient der langfristigen Pufferung der Echtzeituhr. Es kann in den Steckplatz des Signalboards der S7-1200 CPU (ab Firmware 3.0) gesteckt werden. Sie müssen das BB 1297 in die Gerätekonfiguration aufnehmen und die Hardwarekonfiguration in die CPU laden, damit das BB funktionsfähig ist.

Die Batterie (Typ CR1025) ist nicht im Lieferumfang des BB 1297 enthalten und ist separat zu erwerben.

---

#### Hinweis

Das BB 1297 ist mechanisch für die CPUs mit Firmware ab Version 3.0 ausgelegt.

Verwenden Sie das BB 1297 nicht mit CPUs früherer Versionen, weil sich der Stecker des BB 1297 nicht in die CPU stecken lässt.

---

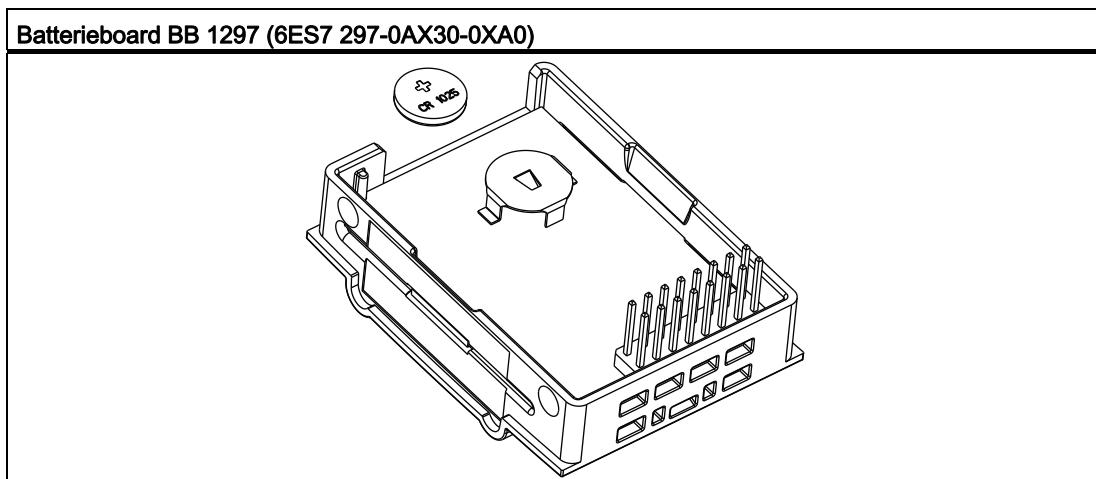
Tabelle A- 209 Allgemeine technische Daten

Technische Daten	BB 1297 Batterieboard
Bestellnummer	6ES7 297-0AX30-0XA0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	28 Gramm
Leistungsverlust	0,5 W
Stromaufnahme (SM-Bus)	11 mA
Stromaufnahme (24 V DC)	Keine

Batterie (nicht enthalten)	BB 1297 Batterieboard
Netzausfallüberbrückung	Ca. 1 Jahr
Batterietyp	CR1025 Siehe Einsetzen oder Austauschen der Batterie des Batterieboards BB 1297 (Seite 55)
Nennspannung	3 V
Nennkapazität	Mindestens 30 mAH

Diagnose	BB 1297 Batterieboard
Kritischer Batteriezustand	< 2,5 V
Batteriediagnose	Anzeige bei geringer Spannung: <ul style="list-style-type: none"> <li>• Geringe Batteriespannung bewirkt, dass die MAINT-LED der CPU ständig bernsteinfarben leuchtet.</li> <li>• Diagnosepufferereignis: 16#06:2700 "Submodulwartung erforderlich: mindestens eine Batterie ist erschöpft (BATTFF)"</li> </ul>
Batteriezustand	Batteriezustandsbit 0 = Batterie OK 1 = Batterie schwach
Aktualisierung des Batteriezustands	Der Batteriezustand wird beim Anlauf und dann einmal täglich aktualisiert, wenn sich die CPU im Betriebszustand RUN befindet.

Tabelle A- 210 Einbauabbildung für das Batterieboard BB 1297



## A.12 Kommunikationsschnittstellen

### A.12.1 PROFIBUS

#### A.12.1.1 CM 1242-5

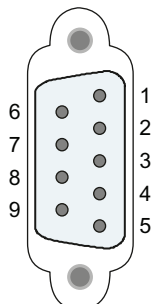
Tabelle A- 211 Technische Daten des CM 1242-5

<b>Technische Daten</b>	
Bestellnummer	6GK7 242-5DX30-0XE0
<b>Schnittstellen</b>	
Anschluss an PROFIBUS	9-polige Sub-D-Buchse
Maximale Stromaufnahme an der PROFIBUS-Schnittstelle beim Anschluss von Netzkomponenten (beispielsweise optische Netzkomponenten)	15 mA bei 5 V (nur für die Bus-Terminierung) *)
<b>Zulässige Umgebungsbedingungen</b>	
Umgebungstemperatur	
<ul style="list-style-type: none"> <li>• während Lagerung</li> <li>• während Transport</li> <li>• während Betriebsphase bei senkrechter Installation (Hutschiene horizontal)</li> <li>• während Betriebsphase bei waagerechter Installation (Hutschiene vertikal)</li> </ul>	<ul style="list-style-type: none"> <li>• -40 °C ... 70 °C</li> <li>• -40 °C ... 70 °C</li> <li>• 0 °C ... 55 °C</li> <li>• 0 °C ... 45 °C</li> </ul>
Relative Luftfeuchte bei 25 °C während der Betriebsphase, ohne Kondensation, maximal	95 %
Schutzart	IP20
<b>Spannungsversorgung, Stromaufnahme, Verlustleistung</b>	
Art der Spannungsversorgung	DC
Spannungsversorgung aus dem Rückwandbus	5 V
Aufgenommener Strom (typisch)	150 mA
Verlustwirkleistung (typisch)	0,75 W
<b>Maße und Gewichte</b>	
<ul style="list-style-type: none"> <li>• Breite</li> <li>• Höhe</li> <li>• Tiefe</li> </ul>	<ul style="list-style-type: none"> <li>• 30 mm</li> <li>• 100 mm</li> <li>• 75 mm</li> </ul>
Gewicht	
<ul style="list-style-type: none"> <li>• Nettogewicht</li> <li>• Gewicht inklusive Verpackung</li> </ul>	<ul style="list-style-type: none"> <li>• 115 g</li> <li>• 152 g</li> </ul>

\*) Die Strombelastung durch einen externen Verbraucher, der zwischen VP (Pin 6) und DGND (Pin 5) angeschlossen wird, darf für die Bus-Terminierung maximal 15 mA betragen (kurzschlussfest).

**PROFIBUS-Schnittstelle**

Tabelle A- 212 Kontaktbelegung der Sub-D-Buchse



Pin	Beschreibung	Pin	Beschreibung
1	- nicht belegt -	6	P5V2: Spannungsversorgung +5V
2	- nicht belegt -	7	- nicht belegt -
3	RxD/TxD-P: Datenader B	8	RxD/TxD-N: Datenader A
4	RTS	9	- nicht belegt -
5	M5V2: Datenbezugspotenzial (Masse DGND)	Gehäuse	Erdungsanschluss

**A.12.1.2 CM 1243-5**

Tabelle A- 213 Technische Daten des CM 1243-5

<b>Technische Daten</b>	
Bestellnummer	6GK7 243-5DX30-0XE0
<b>Schnittstellen</b>	
Anschluss an PROFIBUS	9-polige Sub-D-Buchse
Maximale Stromaufnahme an der PROFIBUS-Schnittstelle beim Anschluss von Netzkomponenten (beispielsweise optische Netzkomponenten)	15 mA bei 5 V (nur für die Bus-Terminierung *)
<b>Zulässige Umgebungsbedingungen</b>	
Umgebungstemperatur	
<ul style="list-style-type: none"> <li>während Lagerung</li> <li>während Transport</li> <li>während Betriebsphase bei senkrechter Installation (Hutschiene horizontal)</li> <li>während Betriebsphase bei waagerechter Installation (Hutschiene vertikal)</li> </ul>	<ul style="list-style-type: none"> <li>-40 °C ... 70 °C</li> <li>-40 °C ... 70 °C</li> <li>0 °C ... 55 °C</li> <li>0 °C ... 45 °C</li> </ul>
Relative Luftfeuchte bei 25 °C während der Betriebsphase, ohne Kondensation, maximal	95 %
Schutzart	IP20
<b>Spannungsversorgung, Stromaufnahme, Verlustleistung</b>	
Art der Spannungsversorgung	DC
Versorgungsspannung / extern	24 V
<ul style="list-style-type: none"> <li>minimal</li> <li>maximal</li> </ul>	<ul style="list-style-type: none"> <li>19,2 V</li> <li>28,8 V</li> </ul>
Aufgenommener Strom (typisch)	
<ul style="list-style-type: none"> <li>aus DC 24 V</li> <li>aus dem S7-1200-Rückwandbus</li> </ul>	<ul style="list-style-type: none"> <li>100 mA</li> <li>0 mA</li> </ul>

### Technische Daten

Verlustwirkleistung (typisch)

- aus DC 24 V • 2,4 W
- aus dem S7-1200-Rückwandbus • 0 W

Spannungsversorgung DC 24 V / extern

- Min. Leitungsquerschnitt • min.: 0,14 mm<sup>2</sup> (AWG 25)
- Max. Leitungsquerschnitt • max.: 1,5 mm<sup>2</sup> (AWG 15)
- Anzugsmoment der Schraubklemmen • 0,45 Nm (4 lb.in.)

### Maße und Gewichte

- Breite • 30 mm
- Höhe • 100 mm
- Tiefe • 75 mm

Gewicht

- Nettogewicht • 134 g
- Gewicht inklusive Verpackung • 171 g

\*) Die Strombelastung durch einen externen Verbraucher, der zwischen VP (Pin 6) und DGND (Pin 5) angeschlossen wird, darf für die Bus-Terminierung maximal 15 mA betragen (kurzschlussfest).

### Hinweis

Das CM 1243-5 (PROFIBUS-Mastermodul) muss über die 24-V-DC-Geberversorgung der CPU gespeist werden.

## PROFIBUS-Schnittstelle

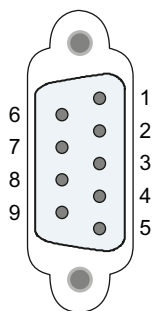


Tabelle A- 214 Kontaktbelegung der Sub-D-Buchse

Pin	Beschreibung	Pin	Beschreibung
1	- nicht belegt -	6	VP: Spannungsversorgung +5V nur für Busabschlusswiderstände (Terminierung); nicht für Versorgung externer Geräte
2	- nicht belegt -	7	- nicht belegt -
3	RxD/TxD-P: Datenader B	8	RxD/TxD-N: Datenader A
4	CNTR-P: RTS	9	- nicht belegt -
5	DGND: Masse für Datensignale und VP	Gehäuse	Erdungsanschluss

**PROFIBUS-Kabel**

**ACHTUNG**

**Auflegen der Schirmung des PROFIBUS-Kabels**

Der Schirm des PROFIBUS-Kabels muss aufgelegt werden.

Isolieren Sie hierzu das PROFIBUS-Kabel am Ende ein Stück ab und verbinden Sie den Schirm mit der Funktionserdung.

**A.12.2 GPRS**

**Hinweis**

**Der GPRS CP hat keine Zulassung für Anwendungen für das Seewesen**

Das folgende Modul hat keine Zulassung für das Seewesen:

- CP 1242-7 GPRS-Modul

**Hinweis**

Um diese Module nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

**A.12.2.1 CP 1242-7**

Tabelle A- 215 Technische Daten des CP 1242-7

<b>Technische Daten</b>	
Bestellnummer	6GK7 242-7KX30-0XE0
<b>Funk-Schnittstelle</b>	
Antennenanschluss	SMA-Buchse
Impedanz nominal	50 Ohm
<b>Funkverbindung</b>	
Maximale Sendeleistung	<ul style="list-style-type: none"> <li>• GSM 850, Class 4: +33 dBm ±2dBm</li> <li>• GSM 900, Class 4: +33 dBm ±2dBm</li> <li>• GSM 1800, Class 1: +30 dBm ±2dBm</li> <li>• GSM 1900, Class 1: +30 dBm ±2dBm</li> </ul>
GPRS	Multislot-Klasse 10 Endgeräteklasse B Kodierungsschema 1...4 (GMSK)
SMS	Betriebsmodus abgehend: MO Dienst: Punkt zu Punkt



---

## Technische Daten

---

### Zulässige Umgebungsbedingungen

---

#### Umgebungstemperatur

- |   |                    |
|---|--------------------|
| • während Lagerung  | • -40 °C ... 70 °C |
| • während Transport   | • -40 °C ... 70 °C |
| • während Betriebsphase bei senkrechter Installation<br>(Hutschiene horizontal) | • 0 °C ... 55 °C   |
| • während Betriebsphase bei waagerechter Installation<br>(Hutschiene vertikal)  | • 0 °C ... 45 °C   |
- 

Relative Luftfeuchte bei 25 °C während der Betriebsphase,  
ohne Kondensation, maximal 95 %

---

Schutzart IP20

---

### Spannungsversorgung, Stromaufnahme, Verlustleistung

---

Art der Spannungsversorgung DC

---

Versorgungsspannung / extern 24 V

- |           |          |
|-----------|----------|
| • minimal | • 19,2 V |
| • maximal | • 28,8 V |
- 

#### Aufgenommener Strom (typisch)

- |                               |          |
|-------------------------------|----------|
| • aus DC 24 V                 | • 100 mA |
| • aus dem S7-1200-Rückwandbus | • 0 mA   |
- 

#### Verlustwirkleistung (typisch)

- |                               |         |
|-------------------------------|---------|
| • aus DC 24 V                 | • 2,4 W |
| • aus dem S7-1200-Rückwandbus | • 0 W   |
- 

#### Spannungsversorgung DC 24 V

- |                                   |                                       |
|-----------------------------------|---------------------------------------|
| • Min. Leitungsquerschnitt        | • min.: 0,14 mm <sup>2</sup> (AWG 25) |
| • Max. Leitungsquerschnitt        | • max.: 1,5 mm <sup>2</sup> (AWG 15)  |
| • Anzugsmoment der Schraubklemmen | • 0,45 Nm (4 lb.in.)                  |
- 

### Maße und Gewichte

---

- |          |          |
|----------|----------|
| • Breite | • 30 mm  |
| • Höhe   | • 100 mm |
| • Tiefe  | • 75 mm  |
- 

#### Gewicht

- |                                |         |
|--------------------------------|---------|
| • Nettogewicht                 | • 133 g |
| • Gewicht inklusive Verpackung | • 170 g |
-

**Technische Daten der GSM/GPRS-Antenne ANT794-4MR**

<b>ANT794-4MR</b>	
Bestellnummer	6NH9860-1AA00
Mobilfunknetze	GSM / GPRS
Frequenzbereiche	<ul style="list-style-type: none"> <li>• 824...960 MHz (GSM 850, 900)</li> <li>• 1 710...1 880 MHz (GSM 1 800)</li> <li>• 1 900...2 200 MHz (GSM / UMTS)</li> </ul>
Charakteristik	omnidirektional
Antennengewinn	0 dB
Impedanz	50 Ohm
Stehwellenverhältnis (SWR)	< 2,0
Max. Leistung	20 W
Polarität	linear vertikal
Stecker	SMA
Länge Antennenkabel	5 m
Außenmaterial	Hart-PVC, UV-beständig
Schutzart	IP20
Zulässige Umgebungsbedingungen	<ul style="list-style-type: none"> <li>• Betriebstemperatur</li> <li>• Transport-/Lagertemperatur</li> <li>• Relative Feuchte</li> </ul>
	<ul style="list-style-type: none"> <li>• -40 °C bis +70 °C</li> <li>• -40 °C bis +70 °C</li> <li>• 100 %</li> </ul>
Außenmaterial	Hart-PVC, UV-beständig
Konstruktiver Aufbau	Antenne mit 5 m fest verbundenem Kabel und SMA-Stecker
Maße (D x H) in mm	25 x 193
Gewicht	<ul style="list-style-type: none"> <li>• Antenne inkl. Kabel</li> <li>• Montageteile</li> </ul>
	<ul style="list-style-type: none"> <li>• 310 g</li> <li>• 54 g</li> </ul>
Montage	Über mitgelieferten Winkel

**Technische Daten der Flachantenne ANT794-3M**

Bestellnummer	6NH9870-1AA00	
Mobilfunknetze	<b>GSM 900</b>	<b>GSM 1800/1900</b>
Frequenzbereiche	890 - 960 MHz	1710 - 1990 MHz
Stehwellenverhältnis (VSWR)	≤ 2:1	≤ 1,5:1
Rückflussdämpfung (Tx)	≈ 10 dB	≈ 14 dB
Antennengewinn	0 dB	
Impedanz	50 Ohm	

Max. Leistung	10 W
Antennenkabel	HF-Kabel RG 174 (fest angeschlossen) mit SMA-Stecker
Kabellänge	1,2 m
Schutzart	IP 64
Zulässiger Temperaturbereich	-40°C bis +75°C
Entflammbarkeit	UL 94 V2
Außenmaterial	ABS Polylac PA-765, lichtgrau (RAL 7035)
Maße (B x L x H) in mm	70,5 x 146,5 x 20,5
Gewicht	130 g

## A.12.3 CM 1243-2 AS-i-Master

### A.12.3.1 Technische Daten des AS-i Master CM 1243-2

Tabelle A- 216 Technische Daten des AS-i Masters CM 1243-2

<b>Technische Daten</b>	
Bestellnummer	3RK7243-2AA30-0XB0
<b>Schnittstellen</b>	
Maximale Stromaufnahme aus SIMATIC Rückwandbus	max. 250 mA, Versorgungsspannung SIMATIC Rückwandbus DC 5 V
aus der AS-i Leitung	max. 100 mA
Pinbelegung	siehe Kapitel Elektrische Anschlüsse des AS-i Masters CM 1243-2 (Seite 868)
Leiterquerschnitt	0,2 mm <sup>2</sup> (AWG 24) ... 3,3 mm <sup>2</sup> (AWG 12)
Anzugsmoment ASI-Stecker	0,56 Nm
<b>Zulässige Umgebungsbedingungen</b>	
Umgebungstemperatur	
während Lagerung	-40 °C ... 70 °C
während Transport	-40 °C ... 70 °C
während Betriebsphase bei senkrechter Installation (Hutschiene horizontal)	0 °C ... 55 °C
während Betriebsphase bei waagerechter Installation (Hutschiene vertikal)	0 °C ... 45 °C
Relative Luftfeuchte bei 25 °C während der Betriebsphase, ohne Kondensation, maximal	95 %
Schutzart	IP20
<b>Spannungsversorgung, Stromaufnahme, Verlustleistung</b>	
Art der Spannungsversorgung	DC
Aufgenommener Strom (typisch) aus dem S7-1200-Rückwandbus	200 mA

Technische Daten	
Verlustwirkleistung (typisch) aus dem S7-1200-Rückwandbus	2,4 W aus AS-i 0,5 W
Maße und Gewichte	
Breite	30 mm
Höhe	100 mm
Tiefe	75 mm
Gewicht	
Nettogewicht	122 g
Gewicht inklusive Verpackung	159 g

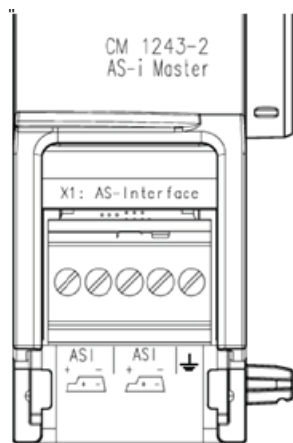
### A.12.3.2 Elektrische Anschlüsse des AS-i Masters CM 1243-2


#### Spannungsversorgung des AS-i Masters CM 1243-2

Der AS-i Master CM 1243-2 wird über den Kommunikationsbus der S7-1200 versorgt. Somit kann auch bei Ausfall der AS-i Versorgungsspannung eine Diagnosemeldung an die S7-1200 gesendet werden. Der Anschluss an den Kommunikationsbus befindet sich rechts am AS-i Master CM 1243-2.

#### Anschlussklemmen AS-Interface

Die abnehmbare Klemme zum Anschluss der AS-i Leitung befindet sich hinter der unteren Abdeckklappe auf der Vorderseite des AS-i Masters CM 1243-2.




Bei Verwendung der AS-i Profilleitung können Sie am Symbol  die richtige Polarität der Anschlussleitung erkennen.

Hinweise zum Ausbau und Wiedereinbau des Klemmenblocks finden Sie im Systemhandbuch "SIMATIC S7-1200 Automatisierungssystem" (Bestellnummer: 6ES7298-8FA30-8AH0).

<b>ACHTUNG</b>
<p><b>Maximale Belastbarkeit der Anschlusskontakte</b></p> <p>Die Belastbarkeit der Anschlusskontakte beträgt max. 8 A. Falls dieser Wert auf der AS-i Leitung überschritten wird, darf der AS-i Master CM 1243-2 nicht in die AS-i Leitung "eingeschleift" werden, sondern muss über eine Stickleitung angeschlossen werden (nur ein Anschlusspaar des AS-i Masters CM 1243-2 belegt).</p> <p>Weitere Hinweise zum Anschluss der AS-i Leitung finden Sie im Kapitel "Montage, Anschluss und Inbetriebnahme der Baugruppen" im Handbuch "AS-i Master CM 1243-2 und AS-i Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200".</p>

### Belegung der Klemmen

<b>Beschriftung</b>	<b>Bedeutung</b>
ASI +	AS-i Anschluss – positive Polarität
ASI –	AS-i Anschluss – negative Polarität
	Funktionserde

## A.12.4 RS232, RS422 und RS485

### A.12.4.1 Technische Daten des CM 1241 RS485

#### Hinweis

Um dieses CB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A- 217 Allgemeine technische Daten

<b>Technische Daten</b>	<b>CB 1241 RS485</b>
Bestellnummer	6ES7 241-1CH30-1XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	40 Gramm

A.12 Kommunikationsschnittstellen

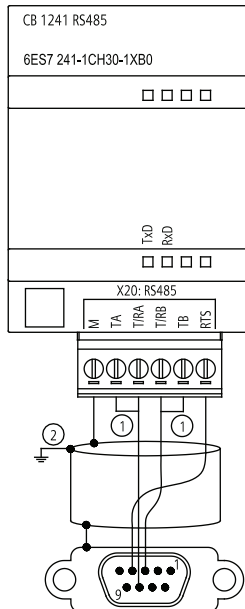
Tabelle A- 218 Sender und Empfänger

Technische Daten	CB 1241 RS485
Typ	RS485 (2-Leiter-Halbduplex)
Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, 3 V fortlaufender Effektivwert
Differentialausgangsspannung Sender	min. 2 V bei $R_L = 100 \Omega$ min. 1,5 V bei $R_L = 54 \Omega$
Abschluss und Bias	10 k zu +5 V an B, RS485 Pin 3 10 k zu GND an A, RS485 Pin 4
Optionaler Abschluss	Kurzschluss Pin TB mit Pin T/RB, effektive Abschlussimpedanz beträgt 127 $\Omega$ , Anschluss an RS485 Pin 3 Kurzschluss Pin TA mit Pin T/RA, effektive Abschlussimpedanz beträgt 127 $\Omega$ , Anschluss an RS485 Pin 4
Eingangsimpedanz Empfänger	min. 5,4 k $\Omega$ einschließlich Abschluss
Ansprechgrenze/Sensibilität Empfänger	min. +/- 0,2 V, 60 mV typ. Hysterese
Potentialtrennung RS485-Signal zu Masse RS485-Signal zu CPU-Logik	500 V AC für 1 Minute
Leitungslänge (geschirmt)	max. 1000 m
Baudrate	300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s (Standard), 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s, 115,2 kBit/s,
Parität	Keine Parität (Standard), gerade, ungerade, Mark (Paritätsbit immer auf 1), Space (Paritätsbit immer auf 0)
Anzahl Stoppbits	1 (Standard), 2
Flusskontrolle	Nicht unterstützt
Wartezeit	0 bis 65535 ms

Tabelle A- 219 Spannungsversorgung

Technische Daten	CB 1241 RS485
Verlustleistung	1,5 W
Max. Stromaufnahme (SM-Bus)	50 mA
Max. Stromaufnahme (24 V DC)	80 mA

CB 1241 RS485 (6ES7 241-1CH30-1XB0)



- ① "TA" und TB" wie gezeigt anschließen, um das Netzwerk abzuschließen. (Nur die Endgeräte im RS485-Netz abschließen.)
- ② Verwenden Sie geschirmte, verdrehte Leiterpaare und schließen Sie den Kabelschirm an Erde an.

Sie schließen nur die zwei Enden des RS485-Netzes ab. Die Geräte zwischen den beiden Endgeräten werden nicht abgeschlossen. Siehe Abschnitt "Abschließen eines RS485-Busanschlusssteckers" im S7-1200 Systemhandbuch.

Tabelle A- 220Anschlussbelegungen für das CB 1241 RS485 (6ES7 241-1CH30-1XB0)

Pin	9-poliger Steckverbinder	X20
1	RS485 / logische Erde	--
2	RS485 / frei	--
3	RS485 / TxD+	3 - T/RB
4	RS485 / RTS	1 - RTS
5	RS485 / logische Erde	--
6	RS485 / 5-V-Leistung	--
7	RS485 / frei	--
8	RS485 / TxD-	4 - T/RA
9	RS485 / frei	--
Hülse		7 - M

**Siehe auch**

Abschließen eines RS485-Busanschlusssteckers (Seite 598)

**A.12.4.2 Technische Daten für CM 1241 RS232**

Tabelle A- 221 Allgemeine technische Daten

Technische Daten	CM 1241 RS232
Bestellnummer	6ES7 241-1AH30-0XB0
Abmessungen (mm)	30 x 100 x 75
Gewicht	150 Gramm

Tabelle A- 222 Sender und Empfänger

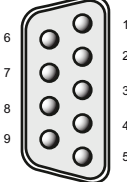
Technische Daten	CM 1241 RS232
Typ	RS232 (Voll duplex)
Ausgangsspannung Sender	min. +/- 5 V bei $R_L = 3\text{ k}\Omega$
Ausgangsspannung Sender	max. +/- 15 V DC
Eingangsimpedanz Empfänger	min. $3\text{ k}\Omega$
Ansprechgrenze/Sensibilität Empfänger	min. 0,8 V Low, max. 2,4 V High, typ. Hysterese 0,5 V
Empfänger-Eingangsspannung	max. +/- 30 V DC
Potentialtrennung RS-232-Signal zu Masse RS-232-Signal zu CPU-Logik	500 V AC für 1 Minute
Leitungslänge (geschirmt)	max. 10 m
Baudrate	300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s (Standard), 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s, 115,2 kBit/s,
Parität	Keine Parität (Standard), gerade, ungerade, Mark (Paritätsbit immer auf 1), Space (Paritätsbit immer auf 0)
Anzahl Stoppbits	1 (Standard), 2
Flusskontrolle	Hardware, Software
Wartezeit	0 bis 65535 ms

Tabelle A- 223 Spannungsversorgung

Technische Daten	CM 1241 RS232
Verlustleistung	1,1 W
aus +5 V DC	220 mA



Tabelle A- 224 RS232-Steckverbinder (Stecker)

Pin	Beschreibung	Steckverbinder (Stecker)	Pin	Beschreibung
1 DCD	Datenträgererkennung: Eingang		6 DSR	Datensatz bereit: Eingang
2 RxD	Daten von DCE empfangen: Eingang		7 RTS	Sendeanforderung: Ausgang
3 TxD	Daten an DCE gesendet: Ausgang		8 CTS	Bereit zum Senden: Eingang
4 DTR	Datenterminal bereit: Ausgang		9 RI	Rufanzeige (nicht verwendet)
5 GND	Logikmasse		SHELL	Erdungsanschluss

### A.12.4.3 Technische Daten des CM 1241 RS422/485

#### Technische Daten des CM 1241 RS422/485

Tabelle A- 225 Allgemeine technische Daten

Technische Daten	CM 1241 RS422/485
Bestellnummer	6ES7 241-1CH31-0XB0
Abmessungen B x H x T (mm)	30 x 100 x 75
Gewicht	155 Gramm

Tabelle A- 226 Sender und Empfänger

Technische Daten	CM 1241 RS422/485
Typ	RS422 oder RS485, 9-polige Sub-D-Buchse
Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, 3 V fortlaufender Effektivwert
Differentialausgangsspannung Sender	min. 2 V bei $R_L = 100 \Omega$ min. 1,5 V bei $R_L = 54 \Omega$
Abschluss und Bias	10 k $\Omega$ zu +5 V an B, PROFIBUS Pin 3 10 k $\Omega$ zu GND an A, PROFIBUS Pin 8 Interne Bias-Optionen vorhanden oder kein interner Bias. In allen Fällen ist der externe Abschluss erforderlich, siehe Abschließen eines RS485-Netzwerksteckers (Seite 598) und Konfigurieren von RS422 und RS485 im S7-1200 Systemhandbuch (Seite 637).
Eingangsimpedanz Empfänger	min. 5,4 k $\Omega$ einschließlich Abschluss
Ansprechgrenze/Sensibilität Empfänger	min. +/- 0,2 V, 60 mV typ. Hysterese
Potentialtrennung RS485-Signal zu Masse RS485-Signal zu CPU-Logik	500 V AC für 1 Minute
Leitungslänge (geschirmt)	max. 1000 m (abhängig von der Baudrate)

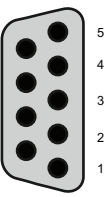
A.13 TeleService (TS-Adapter und TS-Adaptermodul)

Technische Daten	CM 1241 RS422/485
Baudrate	300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s (Standard), 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s, 115,2 kBit/s
Parität	Keine Parität (Standard), gerade, ungerade, Mark (Paritätsbit immer auf 1), Space (Paritätsbit immer auf 0)
Anzahl Stoppbits	1 (Standard), 2
Flusskontrolle	XON/XOFF wird im RS422-Modus unterstützt
Wartezeit	0 bis 65535 ms

Tabelle A- 227 Stromversorgung

Technische Daten	CM 1241 RS422/485
Verlustleistung	1,2 W
aus +5 V DC	240 mA

Tabelle A- 228 RS485- oder RS422-Steckverbinder (Buchse)

Pin	Beschreibung	Steckverbinder (Buchse)	Pin	Beschreibung
1	Logik- oder Kommunikationsmasse		6 PWR	+5 V mit 100 Ohm Reihenwiderstand: Ausgang
2 TxD+ <sup>1</sup>	Angeschlossen für RS422 Nicht belegt für RS485: Ausgang		7	Nicht angeschlossen
3 TxD+	Signal B (RxD/TxD+): Eingang/Ausgang		8 TXD-	Signal A (RxD/TxD-): Eingang/Ausgang
4 RTS <sup>2</sup>	Ausgang Sendeanforderung/RTS (TTL-Pegel)		9 TXD- <sup>1</sup>	Angeschlossen für RS422 Nicht belegt für RS485: Ausgang
5 GND	Logik- oder Kommunikationsmasse		SHELL	Erdungsanschluss

<sup>1</sup> Die Pins 2 und 9 werden lediglich als Sendesignale bei RS422 verwendet.

<sup>2</sup> RTS ist ein Signal auf TTL-Ebene und kann dazu genutzt werden, ein weiteres Halbduplex-Gerät basierend auf diesem Signal zu steuern. Es ist beim Senden aktiv und ansonsten inaktiv.

### A.13 TeleService (TS-Adapter und TS-Adaptermodul)

Die folgenden Handbücher enthalten die technischen Daten des TS-Adapters IE Basic und des TS-Adaptermoduls:

- Industrie-Software Engineering Tools  
Modularer TS-Adapter
- Industrie-Software Engineering Tools  
TS-Adapter IE Basic

## A.14 SIMATIC Memory Cards

<b>Bestellnummer</b>	<b>Kapazität</b>
6ES7 954-8LF01-0AA0	24 MB
6ES7 954-8LE01-0AA0	12 MB
6ES7 954-8LB01-0AA0	2 MB

## A.15 Eingangssimulatoren

Tabelle A- 229 Allgemeine technische Daten

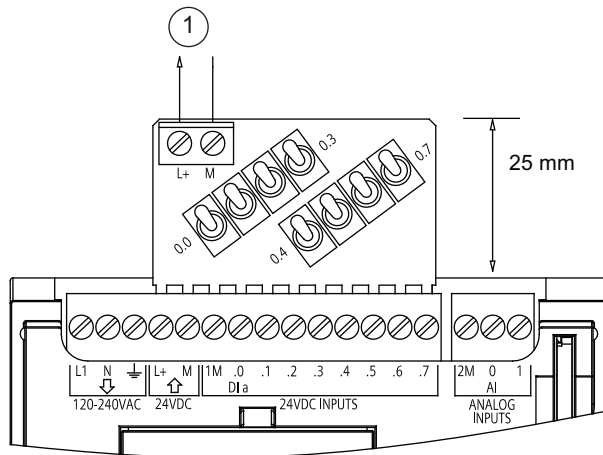
<b>Technische Daten</b>	<b>Simulator mit 8 Anschlussklemmen</b>	<b>Simulator mit 14 Anschlussklemmen</b>
Bestellnummer	6ES7 274-1XF30-0XA0	6ES7 274-1XH30-0XA0
Abmessungen (B x H x T) (mm)	43 x 35 x 23	67 x 35 x 23
Gewicht	20 Gramm	30 Gramm
Ein- und Ausgänge	8	14
Verwendung mit	CPU 1211C, CPU 1212C	CPU 1214C

 **WARNUNG**

Diese Eingangssimulatoren sind nicht geeignet für die Nutzung in explosionsgefährdeten Betriebsstätten der Gefahrengruppen Klasse I DIV 2 und Klasse I Zone 2. Die Schalter können bei Einsatz an Standorten der Klasse I DIV2 bzw. Klasse I Zone 2 zu Funkenbildung/Explosion führen.

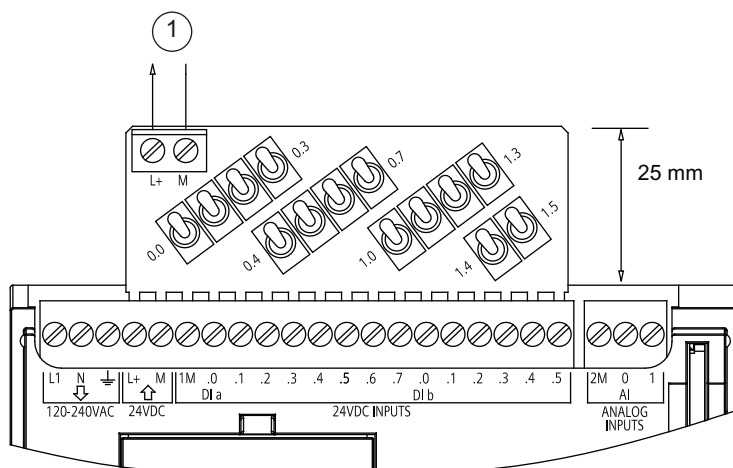
Simulator mit 8 Positionen (6ES7 274-1XF30-0XA0)

① 24-V-DC-Geberspannung



Simulator mit 14 Positionen (6ES7 274-1XF30-0XA0)

① 24-V-DC-Geberspannung



## A.16 Steckleitung für Erweiterungsmodule

---

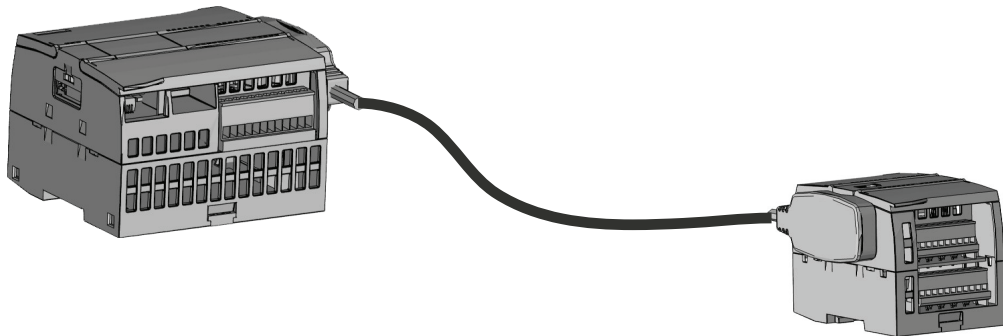
**Technische Daten**

---

Bestellnummer	6ES7 290-6AA30-0XA0
Leitungslänge	2 m
Gewicht	200 g

---

Weitere Informationen zum Einbauen und Ausbauen des S7-1200 Erweiterungskabels finden Sie im Abschnitt "Einbau" (Seite 61).



## A.17 Zugehörige Produkte

### A.17.1 PM 1207 Stromversorgungsmodul

Das PM 1207 ist ein Stromversorgungsmodul für die SIMATIC S7-1200. Es bietet die folgenden Leistungsmerkmale:

- 120/230 V AC Eingang, 24 V DC/2,5 A Ausgang
- Bestellnummer 6ESP 332-1SH71

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Kundensupport-Website (<http://www.siemens.com/automation/>).

### A.17.2 CSM 1277 Compact Switch Module

Das CSM1277 ist ein Compact Switch Module für Industrial Ethernet. Es kann zur Multiplikation der Ethernet-Schnittstelle der S7-1200 eingesetzt werden, um die gleichzeitige Kommunikation mit Bedienpanels, Programmiergeräten oder anderen Steuerungen zu ermöglichen. Das Modul bietet die folgenden Leistungsmerkmale:

- 4 x RJ45-Buchsen für den Anschluss an Industrial Ethernet
- 3-polige Klemmenleiste für den Anschluss der externen 24-V-DC-Versorgung von oben
- LEDs für Diagnose- und Statusanzeige von Industrial Ethernet-Anschlüssen
- Bestellnummer 6GK7 277-1AA00-0AA0

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Kundensupport-Website (<http://www.siemens.com/automation/>).

## Berechnen der Leistungsbilanz

Die CPU besitzt eine interne Spannungsversorgung, die neben der CPU selbst die Erweiterungsmodule und andere 24VDC-Verbraucher versorgt.

Es gibt vier Arten von Erweiterungsmodulen:

- Signalmodule (SM) werden an der rechten Seite der CPU angeschlossen. Für jede CPU gibt es eine maximal anschließbare Anzahl Signalmodule, unabhängig von der Leistungsbilanz.
  - An die CPU 1214C und CPU 1215C können 8 Signalmodule angeschlossen werden.
  - An die CPU 1212C können 2 Signalmodule angeschlossen werden.
  - An die CPU 1211C können keine Signalmodule angeschlossen werden.
- Kommunikationsmodule (CM) werden an der linken Seite der CPU angeschlossen. An jede CPU können maximal 3 Kommunikationsmodule angeschlossen werden, unabhängig von der Leistungsbilanz.
- Signalboards (SB), Kommunikationsboards (CB) und Batterieboards (BB) werden oben auf der CPU angeschlossen. An jede CPU kann maximal 1 Signalboard, Kommunikationsboard oder Batterieboard angeschlossen werden.

Mit Hilfe der folgenden Informationen können Sie berechnen, wieviel Leistung die CPU für Ihre Konfiguration zur Verfügung stellen kann.

Jede CPU liefert Gleichspannung von 5 V und 24 V:

- Die CPU liefert 5V Gleichspannung für eventuell angeschlossene Erweiterungsmodule. Ist der Leistungsbedarf der Erweiterungsmodule an 5V Gleichspannung höher als die interne Spannungsversorgung liefern kann, müssen Sie auf Erweiterungsmodule verzichten, so dass die Leistungsbilanz der S7200 wieder eingehalten wird.
- Jede CPU besitzt eine 24VDC Geberversorgung, die die integrierten Eingänge und die Relaispulen der Erweiterungsmodule mit 24V Gleichspannung versorgt. Ist der Leistungsbedarf der CPU an 24V Gleichspannung höher als die interne Spannungsversorgung liefern kann, können Sie zusätzlich eine externe 24VDC-Versorgung anschließen, die die Erweiterungsmodule speist. Sie müssen die 24VDC-Versorgung manuell an die Eingänge und die Relaispulen anschließen.

 **WARNUNG**

Wenn Sie parallel zu der DC Geberversorgung eine externe 24VDC Spannungsquelle anschließen, kann es sein, dass die beiden Spannungsquellen sich beim Aufbauen der geeigneten Ausgangsspannung gegenseitig beeinträchtigen.

Als Folge kann sich die Lebensdauer verkürzen bzw. eine oder beide Spannungsquellen können sofort ausfallen. Unvorhersehbarer Betrieb kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Die DC-Geberversorgung der CPU und eine externe Spannungsquelle müssen die Spannung an unterschiedlichen Punkten liefern. Dabei darf maximal eine Verbindung zwischen den beiden Spannungsquellen bestehen.

Einige der 24-V-Eingangsports des PLC-Systems sind miteinander verbunden, wobei ein logischer Bezugsleiter mehrere M-Klemmen verbindet. Der 24-V-Versorgungseingang der CPU, der Relaispuleneingang des SM und ein nicht potentialgetrennter Versorgungseingang sind Beispiele für Stromkreise, die miteinander verbunden sind, sofern sie in den Datenblättern als nicht potentialgetrennt angegeben sind. Alle nicht potentialgetrennten M-Klemmen müssen an dasselbe externe Bezugspotential angeschlossen werden.

 **WARNUNG**

Wenn Sie nicht potentialgetrennte M-Klemmen an verschiedene Bezugspotentiale anschließen, verursacht dies unbeabsichtigten Stromfluss, der zu Beschädigung oder unvorhersehbarem Betrieb des Zielsystems und angeschlossener Geräte führen kann.

Beschädigung oder unvorhersehbarer Betrieb können zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Schließen Sie stets alle nicht potentialgetrennten M-Klemmen in einem PLC-System an dasselbe Bezugspotential an.



Informationen zu den Leistungsbilanzen der CPUs und dem Leistungsbedarf der Signalmodule finden Sie in den technischen Daten (Seite 743).

#### Hinweis

Wird die Leistungsbilanz der CPU überschritten, können Sie evtl. nicht die maximale Anzahl Module an Ihre CPU anschließen.

### Beispiel für eine Leistungsbilanz

Im folgenden Beispiel wird der Leistungsbedarf berechnet für eine Konfiguration mit einer CPU 1214C AC/DC/Relais, einem SB 1223 2 x 24 VDC Eingabe/ 2 x 24 VDC Ausgabe, einem CM 1241, drei SM 1223 8 DC-Eing./8 Relaisausg. und einem SM 1221 8 DC Eingabe. Dieses Beispiel umfasst insgesamt 48 Eingänge und 36 Ausgänge.

#### Hinweis

Die CPU hat den für die internen Relaispulen benötigten Leistungsbedarf bereits zugeordnet. Sie brauchen den Leistungsbedarf der internen Relaispulen nicht in die Berechnung der Leistungsbilanz aufzunehmen.

Die CPU in diesem Beispiel liefert genügend 5-V-DC-Spannung für die Signalmodule, doch die Geberversorgung liefert nicht genügend 24-V-DC-Spannung für alle Eingänge und Erweiterungsrelais. Die Ein- und Ausgänge benötigen 456 mA und die CPU liefert nur 400 mA. Dieser Aufbau benötigt zusätzlich mindestens 56 mA an 24-V-DC-Spannung, um alle 24-V-DC-Ein- und -Ausgänge zu versorgen.

Tabelle B- 1 Beispiel für eine Leistungsbilanz

Leistungsbilanz der CPU	5 V DC	24 V DC
CPU 1214C AC/DC/Relais	1600 mA	400 mA
<i>Minus</i>		
Systemvoraussetzungen	5 V DC	24 V DC
CPU 1214C, 14 Eingänge	-	14 * 4 mA = 56 mA
1 SB 1223 2 x 24 VDC Eingabe/ 2 x 24 VDC Ausgabe	50 mA	2 * 4 mA = 8 mA
1 CM 1241 RS422/485, 5 V	220 mA	
3 SM 1223, 5-V-Spannung	3 * 145 mA = 435 mA	-
1 SM 1221, 5-V-Spannung	1 * 105 mA = 105 mA	-
3 SM 1223, je 8 Eingänge	-	3 * 8 * 4 mA = 96 mA
3 SM 1223, je 8 Relaisausgänge	-	3 * 8 * 11 mA = 264 mA
1 SM 1221, je 8 Eingänge	-	8 * 4 mA = 32 mA
<b>Gesamtbedarf</b>	810 mA	456 mA
<i>Gleich</i>		
Leistungsbilanz	5 V DC	24 V DC
GesamtLeistungsbilanz	790 mA	(56 mA)

### Formular zum Berechnen der Leistungsbilanz

Mit Hilfe dieser Tabelle können Sie berechnen, wieviel Leistung die S7-1200 CPU für Ihre Konfiguration liefern kann. Die technischen Daten (Seite 743) bieten Informationen zu der verfügbaren Leistung der einzelnen Ausführungen der CPUs und zum Leistungsbedarf Ihrer Signalmodule.

Tabelle B- 2 Berechnen der Leistungsbilanz

Leistungsbilanz der CPU	5 V DC	24 V DC
<i>Minus</i>		
Systemvoraussetzungen	5 V DC	24 V DC
<b>Gesamtbedarf</b>		
<i>Gleich</i>		
Leistungsbilanz	5 V DC	24 V DC
GesamtLeistungsbilanz		

# Bestellnummern



## C.1 CPU-Module

Tabelle C- 1 S71200 CPUs

CPU-Varianten		Bestellnummer
CPU 1211C	CPU 1211C DC/DC/DC	6ES7 211-1AE31-0XB0
	CPU 1211C AC/DC/Relais	6ES7 211-1BE31-0XB0
	CPU 1211C DC/DC/Relais	6ES7 211-1HE31-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7 212-1AE31-0XB0
	CPU 1212C AC/DC/Relais	6ES7 212-1BE31-0XB0
	CPU 1212C DC/DC/Relais	6ES7 212-1HE31-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7 214-1AG31-0XB0
	CPU 1214C AC/DC/Relais	6ES7 214-1BG31-0XB0
	CPU 1214C DC/DC/Relais	6ES7 214-1HG31-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7 215-1AG31-0XB0
	CPU 1215C AC/DC/Relais	6ES7 215-1BG31-0XB0
	CPU 1215C DC/DC/Relais	6ES7 215-1HG31-0XB0

## C.2 Signalmodule (SMs), Signalboards (SBs) und Batterieboards (BBs)

Tabelle C- 2 Signalmodule (SMs)

Signalmodule		Bestellnummer
Digitaleingang	SM 1221 8 x 24-V-DC-Eingang (stromziehend/stromliefernd)	6ES7 221-1BF30-0XB0
	SM 1221 16 x 24-V-DC-Eingang (stromziehend/stromliefernd)	6ES7 221-1BH30-0XB0
Digitalausgang	SM 1222 8 x 24-V-DC-Ausgang (stromliefernd)	6ES7 222-1BF30-0XB0
	SM 1222 16 x 24-V-DC-Ausgang (stromliefernd)	6ES7 222-1BH30-0XB0
	SM 1222 8 x Relaisausgang	6ES7 222-1HF30-0XB0
	SM 1222 8 x Relaisausgang (Umschaltung)	6ES7 222-1XF30-0XB0
	SM 1222 16 x Relaisausgang	6ES7 222-1HH30-0XB0
Digitaleingänge/-ausgänge	SM 1223 8 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 8 x 24-V-DC-Ausgang (stromliefernd)	6ES7 223-1BH30-0XB0
	SM 1223 16 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 16 x 24-V-DC-Ausgang (stromliefernd)	6ES7 223-1BL30-0XB0
	SM 1223 8 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 8 x Relaisausgang	6ES7 223-1PH30-0XB0
	SM 1223 16 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 16 x Relaisausgang	6ES7 223-1PL30-0XB0

C.2 Signalmodule (SMs), Signalboards (SBs) und Batterieboards (BBs)

Signalmodule		Bestellnummer
	SM 1223 8 x 120/230-V-AC-Eingang (stromziehend/stromliefernd) / 8 x Relaisausgang	6ES7 223-1QH30-0XB0
Analogeingang	SM 1231 4 x Analogeingang	6ES7 231-4HD30-0XB0
	SM 1231 8 x Analogeingang	6ES7 231-4HF30-0XB0
	SM 1231 4 x Analogeingang x 16 Bit (hochfunktionell)	6ES7 231-5ND30-0XB0
Analogausgang	SM 1232 2 x Analogausgang	6ES7 232-4HB30-0XB0
	SM 1232 4 x Analogausgang	6ES7 232-4HD30-0XB0
Analogeingänge/-ausgänge	SM 1234 4 x Analogeingang / 2 x Analogausgang	6ES7 234-4HE30-0XB0
RTD und Thermoelement	SM 1231 TC 4 x 16 Bit	6ES7 231-5QD30-0XB0
	SM 1231 TC 8 x 16 Bit	6ES7 231-5QF30-0XB0
	SM 1231 RTD 4 x 16 Bit	6ES7 231-5PD30-0XB0
	SM 1231 RTD 8 x 16 Bit	6ES7 231-5PF30-0XB0

Tabelle C- 3 Signalboards (SBs) und Batterieboards (BBs)

Signal- und Batterieboards		Bestellnummer
Digitaleingang	SB 1221 200 kHz, 4 x 24-V-DC-Eingang (stromliefernd)	6ES7 221-3BD30-0XB0
	SB 1221 200 kHz, 4 x 5-V-DC-Eingang (stromliefernd)	6ES7 221-3AD30-0XB0
Digitalausgang	SB 1222 200 kHz, 4 x 24-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7 222-1BD30-0XB0
	SB 1222 200 kHz, 4 x 5-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7 222-1AD30-0XB0
Digitaleingänge/-ausgänge	SB 1223 2 x 24-V-DC-Eingang (stromziehend) / 2 x 24-V-DC-Ausgang (stromliefernd)	6ES7 223-0BD30-0XB0
	SB 1223 200 kHz, 2 x 24-V-DC-Eingang (stromliefernd) / 2 x 24-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7 223-3BD30-0XB0
	SB 1223 200 kHz, 2 x 5-V-DC-Eingang (stromliefernd) / 2 x 5-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7 223-3AD30-0XB0
Analog	SB 1232 1 Analogausgang	6ES7 232-4HA30-0XB0
	SB 1231 1 Analogeingang	6ES7 231-4HA30-0XB0
	SB 1231 1 Analogeingang Thermoelement	6ES7 231-5QA30-0XB0
	SB 1231 1 Analogeingang RTD	6ES7 231-5PA30-0XB0
Batterie	BB 1297 Batterie	6ES7 297-0AX30-0XA0

## C.3 Kommunikation

Tabelle C- 4 Kommunikationsmodul (CM)

Kommunikationsmodul (CM)			Bestellnummer
RS232, RS422 und RS485	CM 1241 RS232	RS232	6ES7 241-1AH30-0XB0
	CM 1241 RS422/485	RS422/485	6ES7 241-1CH31-0XB0
PROFIBUS	CM 1243-5	PROFIBUS-Master	6GK7 243-5DX30-0XE0
	CM 1242-5	PROFIBUS-Slave	6GK7 242-5DX30-0XE0
AS-i-Master	CM 1243-2	AS-i-Master	3RK7 243-2AA30-0XB0

Tabelle C- 5 Kommunikationsboard (CB)

Kommunikationsboard (CB)			Bestellnummer
RS485	CB 1241 RS485	RS485	6ES7 241-1CH30-1XB0

Tabelle C- 6 Kommunikationsprozessor (CP)

Kommunikationsprozessor (CP)		Bestellnummer
CP 1242-7	GPRS	6GK7 242-7KX30-0XE0

Tabelle C- 7 TeleService

TS-Adapter	Bestellnummer
TS-Adapter IE Basic	6ES7 972-0EB00-0XA0
TS-Modul GSM	6GK7 972-0MG00-0XA0
TS-Modul RS232	6ES7 972-0MS00-0XA0
TS-Modul Modem	6ES7 972-0MM00-0XA0
TS-Modul ISDN	6ES7 972-0MD00-0XA0

Tabelle C- 8 Zubehör

Zubehör			Bestellnummer
Antenne	ANT794-4MR	GSM/GPRS-Antenne	6NH9 860-1AA00
	ANT794-3M	Flachantenne	6NH9 870-1AA00

Tabelle C- 9 Steckverbinder

Art des Steckverbinders		Bestellnummer
RS485	Schraubklemmenanschluss, 35-Grad-Kabelabgang	6ES7 972-0BA42-0XA0
	FastConnect-Anschluss, 35-Grad-Kabelabgang	6ES7 972-0BA60-0XA0

## C.4 Sonstige Module

Tabelle C- 10 Zugehörige Produkte

Beschreibung		Bestellnummer
Spannungsversorgungsmodul	PM 1207 Stromversorgung	6EP1 332-1SH71
Ethernet-Switch	CSM 1277 Ethernet-Switch - 4 Ports	6GK7 277-1AA10-0AA0

## C.5 Memory Cards

Tabelle C- 11 Memory Cards

SIMATIC Memory Cards	Bestellnummer
SIMATIC MC 2 MB	6ES7 954-8LB01-0AA0
SIMATIC MC 12 MB	6ES7 954-8LE01-0AA0
SIMATIC MC 24 MB	6ES7 954-8LF01-0AA0

## C.6 Grundlegende HMI-Geräte

Tabelle C- 12 HMI-Geräte

Grundlegende HMI-Panels	Bestellnummer
KTP400 Basic (Mono, PN)	6AV6 647-0AA11-3AX0
KTP600 Basic (Mono, PN)	6AV6 647-0AB11-3AX0
KTP600 Basic (Farbe, PN)	6AV6 647-0AD11-3AX0
KTP1000 Basic (Farbe, PN)	6AV6 647-0AF11-3AX0
TP1500 Basic (Farbe, PN)	6AV6 647-0AG11-3AX0

## C.7 Ersatzteile und sonstige Hardware

Tabelle C- 13 Erweiterungskabel, Simulatoren und Klemmenblöcke

Beschreibung		Bestellnummer	
Steckleitung für Erweiterungsmodule	Steckleitung für Erweiterungsmodule, 2 m	6ES7 290-6AA30-0XA0	
E/A-Simulator	Simulator (1214C/1211C - 8 E/A)	6ES7 274-1XF30-0XA0	
	Simulator (1214C - 14 E/A)	6ES7 274-1XH30-0XA0	
Ersatzabdeckklappe	CPU 1211C/1212C	6ES7 291-1AA30-0XA0	
	CPU 1214C	6ES7 291-1AB30-0XA0	
	CPU 1215C	6ES7 291-1AC30-0XA0	
	Signalmodul (SM), 45 mm	6ES7 291-1BA30-0XA0	
	Signalmodul (SM), 70 mm	6ES7 291-1BB30-0XA0	
	Kommunikationsmodul	6ES7 291-1CC30-0XA0	
Klemmenblock	Verzinkt	7 Klemmen, 4/Packung	6ES7 292-1AG30-0XA0
		8 Klemmen, 4/Packung	6ES7 292-1AH30-0XA0
		11 Klemmen, 4/Packung	6ES7 292-1AL30-0XA0
		12 Klemmen, 4/Packung	6ES7 292-1AM30-0XA0
		14 Klemmen, 4/Packung	6ES7 292-1AP30-0XA0
		20 Klemmen, 4/Packung	6ES7 292-1AV30-0XA0
	Vergoldet	3 Klemmen, 4/Packung (für Analog-CPUs)	6ES7 292-1BC30-0XA0
		6 Klemmen, 4/Packung (für Signalboards)	6ES7 292-1BF30-0XA0
		6 Klemmen, 4/Packung (für Analog-CPUs)	6ES7 292-1BF30-0XB0
		7 Klemmen, 4/Packung (für Analog-Signalmodule)	6ES7 292-1BG30-0XA0
		11 Klemmen, 4/Packung (für Analog-Signalmodule)	6ES7 292-1BL30-0XA0

## C.8 Programmiersoftware

Tabelle C- 14 Programmiersoftware

SIMATIC-Software		Bestellnummer
Programmiersoftware	STEP 7 Basic V11	6ES7 822-0AA01-0YA0
	STEP 7 Professional V11	6ES7 822-1AA01-0YA5
Visualisierungssoftware	WinCC Basic V11	6AV2100-0AA01-0AA0
	WinCC Comfort V11	6AV2101-0AA01-0AA5
	WinCC Advanced V11	6AV2102-0AA01-0AA5
	WinCC Professional 512 PowerTags V11	6AV2103-0DA01-0AA5
	WinCC Professional 4096 PowerTags V11	6AV2103-0HA01-0AA5
	WinCC Professional max. PowerTags V11	6AV2103-0XA01-0AA5

## C.9 Dokumentation

Tabelle C- 15 S7-1200 Dokumentation

Gedruckte Dokumentation	Sprache	Bestellnummer
Automatisierungssystem S7-1200 Systemhandbuch	Deutsch	6ES7 298-8FA30-8AH0
	Englisch	6ES7 298-8FA30-8BH0
	Französisch	6ES7 298-8FA30-8CH0
	Spanisch	6ES7 298-8FA30-8DH0
	Italienisch	6ES7 298-8FA30-8EH0
	Chinesisch	6ES7 298-8FA30-8KH0
S7-1200 Easy Book	Deutsch	6ES7 298-8FA30-8AQ0
	Englisch	6ES7 298-8FA30-8BQ0
	Französisch	6ES7 298-8FA30-8CQ0
	Spanisch	6ES7 298-8FA30-8DQ0
	Italienisch	6ES7 298-8FA30-8EQ0
	Chinesisch	6ES7 298-8FA30-8KQ0



# Index

## A

- Abfragearchitektur, 632
- Abfragearchitektur Master, 632
- Abfragearchitektur Slave, 632
- ABS (Absolutwert bilden), 214
- Abstand
  - Luftströmung und Kühlung, 46
- AC
  - Erdung, 69
  - Induktive Lasten, 71
  - Richtlinien für Potentialtrennung, 68
  - Verdrahtungsrichtlinien, 67, 69
- ACOS (Arcuscosinus oder inverser Cosinus), 216
- ADD (Addieren), 211
- Ad-hoc-Modus
  - ISO on TCP, 458
  - TCP, 458
- Adressen im Speicher, 96, 97
- Adressierung
  - Boolesche Werte oder Bitwerte, 97
  - Einzelne Eingänge (E) oder Ausgänge (A), 97
  - Prozessabbild, 96
  - Speicherbereiche, 96
- Aktive/passive Kommunikation
  - Parameter, 140
  - Partner konfigurieren, 138, 523
  - Verbindungs-IDs, 454
- Aktive/passive Verbindung, 451
- Alarm
  - Peripheriezugriff, 499
- Alarmbearbeitung, 314
- Alarmer
  - Alarmlatenz, 81
  - ATTACH und DETACH, 305
  - CAN\_DINT (Verzögerungsalarm stornieren), 312
  - SRT\_DINT (Verzögerungsalarm starten), 312
  - Übersicht, 80
- Aliasnamen in benutzerdefinierten Webseiten, 560
- Allgemeine technische Daten, 743
- Analoge E/A
  - Darstellung Ausgang (Spannung), 819, 849
  - Darstellung Ausgang (Strom), 819, 849
  - Darstellung Eingang (Spannung), 818, 848
  - Darstellung Eingang (Strom), 818, 848
  - Konfiguration, 136
  - Schrittantwortzeiten (CPU), 756, 766, 776, 786
  - Schrittantwortzeiten (SB), 847
  - Schrittantwortzeiten (SM), 817
  - Statusanzeigen, 718
  - Umwandlung in physikalische Einheiten, 36, 101, 232
- Analoges Signalboard (SB)
  - SB 1231 AI 1 x 12 Bit, 843
  - SB 1231 AI 1 x 16 Bit RTD, 854
  - SB 1231 AI 1 x 16 Bit Thermoelement, 850
  - SB 1232 AO 1x12 Bit, 845
- Analoges Signalmodul (SM)
  - SM 1231 AI 4 x RTD x 16 Bit, 826
  - SM 1231 AI 8 x 13 Bit, 808
  - SM 1231 AI 8 x 16 Bit TC, 820
  - SM 1231 AI 8 x RTD x 16 Bit, 826
  - SM 1231 AI4 x 16 Bit TC, 820
  - SM 1232 AO 2 x 14 Bit, 812
  - SM 1232 AO 4 x 14 Bit, 812
  - SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 814
- Analogwerte normalisieren, 232
- Analogwerte skalieren, 36, 232
- Ändern
  - Beobachtungstabelle, 729
  - Zustand im Programmiereditor, 728
- Anfangswerte
  - Startwerte eines DBs erfassen und zurücksetzen, 728
- Anlauf nach NETZ-EIN, 75
  - Anlaufverarbeitung, 78
- Anlauf-OB, 80
- Anlaufparameter, 119
- Anmelden/Abmelden
  - Standard-Webseiten, 536
- Anschlüsse
  - Anzahl der Verbindungen (PROFINET), 451
  - Ethernet-Protokolle, 522
  - Kommunikationsarten, 449
  - Konfiguration, 140
  - Partner, 138, 523
  - S7-Verbindung, 522
  - Typen, Verbindungen mit mehreren Teilnehmern, 522
  - Verbindungs-IDs, 454
- Anschlusskontakte
  - Maximale Belastbarkeit, 869
- Anweisung NOT\_OK, 208
- Anweisung OK, 208

- Anweisung TRCV\_C konfigurieren, 495
- Anweisung TSEND\_C konfigurieren, 494
- Anweisungen
  - ABS (Absolutwert bilden), 214
  - ACOS (Arcuscosinus oder inverser Cosinus), 216
  - ADD (Addieren), 211
  - Analogwerte skalieren, 36
  - ASIN (Arcussinus oder inverser Sinus), 216
  - ATAN (Arcustangens oder inverser Tangens), 216
  - ATH (ASCII-Zeichenkette in Hexadezimalzahl wandeln), 280
  - ATTACH, 305
  - Beobachten, 727, 728
  - Bereich kopieren (MOVE\_BLK), 218
  - Bereich ununterbrechbar kopieren (UMOVE\_BLK), 218
  - Betriebsstundenzähler, 265
  - Bitverknüpfung, 185
  - CALCULATE, 35, 209
  - CAN\_DINT (Verzögerungsalarm stornieren), 312
  - CASE (SCL), 236
  - CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 230
  - CONCAT (Verketten), 283
  - CONTINUE (SCL), 240
  - CONV (Umwandeln), 225
  - COS (Cosinus), 216
  - CTD (Rückwärtszählen), 201
  - CTRL\_PWM, 328
  - CTU (Vorwärtszählen), 201
  - CTUD (Vorwärts- und Rückwärtszählen), 201
  - DataLogClose (Datenprotokoll schließen), 340
  - DataLogCreate (Datenprotokoll erstellen), 335
  - DataLogNewFile (Datenprotokoll basierend auf vorhandenem Datenprotokoll erstellen), 342
  - DataLogOpen (Datenprotokoll öffnen), 338
  - DataLogWrite (Datenprotokoll schreiben), 341
  - Datenbausteinststeuerung, 353
  - Datum, 261
  - DEC (Dekrementieren), 213
  - DECO (Decodieren), 254
  - DELETE (Teilzeichenkette löschen), 285
  - DETACH, 305
  - DeviceStates, 318
  - Dezentrale Peripherie AS-i, 289
  - Dezentrale Peripherie PROFIBUS, 289
  - Dezentrale Peripherie PROFINET, 289
  - DIS\_AIRT (Alarmbearbeitung deaktivieren), 314
  - DIV (Dividieren), 211
  - DPNRM\_DG, 302
  - DPRD\_DAT, 300
  - DPWR\_DAT, 300
  - Drag & Drop, 33
  - Drag & Drop zwischen Editoren, 39
  - Einfügen, 33
  - EN\_AIRT (Alarmbearbeitung aktivieren), 314
  - ENCO (Encodieren), 254
  - Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 37
  - Erweiterbare Anweisungen, 37
  - EXIT (SCL), 241
  - EXP (Exponentialwert), 216
  - EXPT (Potenzieren), 216
  - Fallende Flanke, 191
  - Favoriten, 34
  - FeldLesen, 221
  - FeldSchreiben, 221
  - FILL\_BLK, 223
  - FIND (Teilzeichenkette finden), 288
  - FLOOR, 230
  - FOR (SCL), 237
  - Forcefunktion, 733
  - Forcen, 732
  - FRAC (Nachkommastellen), 216
  - Funktionsweise der Zeiten, 197
  - Gemeinsame Parameter, 485
  - GET, 518
  - GET\_DIAG, 321
  - GetError, 249
  - GetErrorID, 250
  - Gleitpunktarithmetik, 216
  - GOTO (SCL), 241
  - Grenzwert, 216
  - HSC (schneller Zähler), 357, 359
  - HTA (Hex in ASCII), 281
  - IF-THEN (SCL), 235
  - IN\_RANGE (innerhalb eines Bereichs), 208
  - INC (Inkrementieren), 213
  - INSERT (Teilzeichenkette einfügen), 286
  - INV (Invertieren), 253
  - JMP, 242
  - JMP\_LIST, 243
  - Kalender, 261
  - LED-Zustand, 316
  - LEFT (Linke Teilzeichenkette), 284
  - LEN (Länge), 282
  - LN (Natürlicher Logarithmus), 216
  - MAX (Maximum), 215
  - MB\_CLIENT, 662
  - MC\_ChangeDynamic, 418
  - MC\_CommandTable, 415
  - MC\_Halt, 404
  - MC\_Home, 401
  - MC\_MoveAbsolute, 406

- MC\_MoveJog, 413  
 MC\_MoveRelative, 408  
 MC\_MoveVelocity, 410  
 MC\_Power, 397  
 MC\_Reset, 400  
 MID (Mittlere Teilzeichenkette), 284  
 MIN (Minimum), 215  
 MOD (Modulo), 212  
 ModuleStates, 319  
 MOVE, 218  
 MUL (Multiplizieren), 211  
 MUX (Multiplexen), 256  
 N\_TRIG, 192  
 NEG (Negation), 213  
 NORM\_X (Normalisieren), 231  
 NOT\_OK, 208  
 ODER, 252  
 OK, 208  
 OUT\_RANGE (außerhalb eines Bereichs), 208  
 P\_TRIG, 192  
 PID\_Compact, 370  
 PORT\_CFG (Portkonfiguration), 602  
 Programmsteuerung (SCL), 234  
 PUT, 518  
 QRY\_CINT (Weckalarm abfragen), 310  
 RALRM, 293  
 RCV\_CFG (Empfangskonfiguration), 605  
 RCV\_PTP (Punkt-zu-Punkt-Daten empfangen), 614  
 RCV\_RST (Empfänger zurücksetzen), 615  
 RD\_LOC\_T (Lokalzeit lesen), 263  
 RD\_SYS\_T (Systemzeit lesen), 263  
 RDREC, 290  
 RE\_TRIGR, 87, 247  
 REPEAT (SCL), 239  
 REPLACE (Teilzeichenkette ersetzen), 287  
 RETURN (SCL), 242  
 RIGHT (Rechte Teilzeichenkette), 284  
 ROL und ROR (Links rotieren und Rechts rotieren), 259  
 ROUND, 229  
 RT (Zeit zurücksetzen), 193  
 Rückgabewert (RET), 246  
 Rücksetzen, 188  
 S\_CONV (Umwandlungen Zahlenwert in Zeichenkette), 269  
 S\_MOV (Zeichenkette verschieben), 269  
 SCALE\_X (Skalieren), 231  
 SEL (Selektieren), 255  
 SEND\_CFG (Sendekonfiguration), 604  
 SEND\_PTP (Punkt-zu-Punkt-Daten senden), 610  
 SET\_CINT (Weckalarm setzen), 308  
 SET\_TIMEZONE, 267  
 Setzen, 188  
 SGN\_GET (RS232-Signale abrufen), 616  
 SGN\_SET (RS232-Signale setzen), 617  
 SHL und SHR (Links schieben und Rechts schieben), 258  
 SIN (Sinus), 216  
 Spalten und Überschriften, 38, 678  
 Sprungmarke, 242  
 SQR (Quadrat), 216  
 SQRT (Quadratwurzel), 216  
 SRT\_DINT (Verzögerungsalarm starten), 312  
 Steigende Flanke, 191  
 STP (PLC-Zyklus stoppen), 248  
 STRG\_VAL (Zeichenkette in Zahlenwert umwandeln), 269  
 SUB (Subtrahieren), 211  
 SWAP, 224  
 SWITCH, 244  
 T\_ADD (Zeiten addieren), 262  
 T\_COMBINE (Zeiten verknüpfen), 263  
 T\_CONFIG, 479  
 T\_CONV (Zeiten umwandeln), 261  
 T\_DIFF (Zeitdifferenz), 262  
 T\_SUB (Zeiten subtrahieren), 262  
 TAN (Tangens), 216  
 TCON, 466  
 TDISCON, 466  
 TOF (Ausschaltverzögerung), 193  
 TON (Einschaltverzögerung), 193  
 TONR (Speichernde Einschaltverzögerung), 193  
 TP (Impuls), 193  
 TRCV, 466  
 TRCV\_C, 459, 494  
 TRUNC (Ganzzahl erzeugen), 229  
 TSEND, 466  
 TSEND\_C, 459, 493  
 TURCV, 474  
 TUSEND, 474  
 UFILL\_BLK (Bereich ununterbrechbar befüllen), 223  
 Uhr, 263  
 Uhrzeit, 261  
 Umwandlungsanweisungen in SCL, 226  
 UND, 252  
 USS\_DRV, 647  
 USS\_PORT, 650  
 USS\_RPM, 651  
 USS\_WPM, 652  
 USS-Statuscodes, 654  
 VAL\_STRG (Zahlenwert in Zeichenkette wandeln), 269  
 Vergleichen, 207  
 Versionen von Anweisungen, 38, 678

- WHILE (SCL), 238
  - WR\_SYS\_T (Systemzeit schreiben), 263
  - WRREC, 290
  - WWW (benutzerdefinierte Webseiten aktivieren), 569
  - XOR (Exklusiv ODER), 252
  - Zähler, 201
  - Zeit, 193
  - Zustand, 727, 728
  - Anwenderprogramm
    - Bausteine aus einer Online-CPU kopieren, 179
    - Codebausteine im Anwenderprogramm aufrufen, 158
    - Drag & Drop zwischen Editoren, 39
    - Einfügen von Anweisungen, 33
    - Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 37
    - Erweiterbare Anweisungen, 37
    - Favoriten, 34
    - Laden in CPU, 179
    - Lineare und strukturierte Programme, 156
    - Memory Card, 117
    - Mit CPU oder Memory Card verknüpfen, 178
    - Organisationsbaustein (OB), 159
    - Passwortschutz, 177
    - Programmkarte, 117
    - Übertragungskarte, 117
  - Anzeige erster Zyklus, 93
  - Arbeitsspeicher, 20
    - CPU 1211C, 750
    - CPU 1212C, 760
    - CPU 1214C, 770
    - CPU 1215C, 780
  - Arithmetik, 35, 209, 211
  - Arrays
    - Mit einer Variable auf Elemente zugreifen, 222
  - Arrays mit Variablen indexieren, 222
  - ASi
    - Analogwerte übertragen,
    - ASi-Adresse,
    - AS-i-Mastermodul CM1243-2 hinzufügen,
    - AS-i-Slave hinzufügen,
    - Digitalwerte übertragen,
    - Eigenschaften der ASi-Adresse,
    - Netzwerkverbindung,
    - RDREC,
    - Systemzuweisung,
    - Systemzuweisung von Slaveadressen,
    - WRREC,
  - AS-i
    - Anweisungen für die dezentrale Peripherie, 289
  - AS-I
    - Slavekonfiguration mit STEP 7, 514
    - Slavekonfiguration ohne STEP 7, 513
  - ASi-Adresse,
    - Konfigurieren,
  - AS-i-Master CM 1243-2, 509
    - Modulfunktionen, 509
  - ASIN (Arcussinus oder inverser Sinus), 216
  - Assistent für den Zertifikatsimport, 593
  - ATEX-Zulassung, 744
  - ATH (ASCII-Zeichenkette in Hexadezimalzahl wandeln), 280
  - ATTACH, 305
  - Auf Werkseinstellungen zurücksetzen, 722
  - Aufrufstruktur, 183
  - Ausgangsparameter, 161
    - Impulsausgänge, 330
    - Impulskanäle konfigurieren, 332
  - Ausschaltverzögerung (TOF), 193
    - Funktionsweise, 197
  - AWP\_Enum\_Def, 561
  - AWP\_Enum\_Ref, 562
  - AWP\_Import\_Fragment, 564
  - AWP\_In\_Variable, 555, 559
  - AWP\_Out\_Variable, 557
  - AWP\_Start\_Fragment, 563
  - AWP-Befehle, 553
    - Alias verwenden, 560
    - Definitionen verbinden, 565
    - Enum-Typ definieren, 561
    - Enum-Typ referenzieren, 562
    - Fragmente generieren, 563
    - Fragmente importieren, 564
    - Sondervariablen lesen, 557
    - Sondervariablen schreiben, 559
    - Variablen schreiben, 555
- B**
- Baudrate, 620
  - Baustein
    - Konsistenzprüfung, 183
  - Bausteine
    - Alarmer, 21, 81, 751, 761, 771, 782
    - Anfangswert eines FB, 161
    - Anlauf-OBs, 81
    - Anzahl der Codebausteine, 21, 73, 751, 761, 771, 782
    - Anzahl der OBs, 21, 81, 751, 761, 771, 782
    - Bausteinaufrufe, 73
    - Bausteine aus einer Online-CPU kopieren, 179
    - Beobachten, 21, 751, 761, 771, 782
    - Datenbaustein (DB), 73

- Einzelinstanz- oder Multiinstanz-DB, 161
- Ereignisse, 81
- FB oder FC mit SCL aufrufen, 168
- Funktion (FC), 73, 160
- Funktionsbaustein (FB), 73, 161
- Größe des Anwenderprogramms, 21, 73, 751, 761, 771, 782
- Gültige FC-, FB- und DB-Nummern, 73
- Instanz-Datenbaustein (DB), 161
- Laden, 179
- Lineare und strukturierte Programme, 156
- Organisationsbausteine (OBs), 21, 73, 80, 81, 751, 761, 771, 782
- Passwortschutz, 177
- Schachtelungstiefe, 21, 73, 751, 761, 771, 782
- Zähler (Anzahl und Speicherbedarf), 21, 752, 762, 772, 782
- Zeiten (Anzahl und Speicherbedarf), 21, 752, 762, 772, 782
- Bausteine aus einer Online-CPU kopieren, 179
- Bedienoberfläche
  - Portalansicht, 32
  - Projektansicht, 33
- Bedienpanel, 28, 40
  - Betriebszustände der CPU, 75
- Beispiel
  - Benutzerdefinierte Webseiten, 572
  - Modbus-Slave, 697
  - PtP-Kommunikation, 633
  - PtP-Kommunikation, Ausführen, 642
  - PtP-Kommunikation, Konfiguration, 634
  - PtP-Kommunikation, STEP 7-Programmierung, 640
  - PtP-Kommunikation, Terminalemulator, 641
- Bemessungsspannungen, 748
- Benutzerdefinierte Webseiten, 531, 551
  - Aktivieren mit WWW-Anweisung, 569
  - Aktualisieren, 553
  - Anlegen mit HTML-Editor, 552
  - AWP-Befehle für den Zugriff auf S7-1200
  - Daten, 553
  - Beispiel, 572
  - Einschränkungen Ladespeicher, 571
  - Entsprechende DBs in die CPU laden, 571
  - Fragmente erstellen, 563
  - Fragmente importieren, 564
  - Handhabung von Sonderzeichen, 565
  - HTML-Liste, 578
  - Konfiguration für mehrere Sprachen, 587
  - Konfigurieren, 567
  - Manuelle Steuerung von DB-Fragmenten, 587
  - mehrere Sprachen, 583
  - Programmbausteine generieren, 569
  - Programmbausteine löschen, 569
  - Programmieren in STEP 7, 569
  - Sondervariablen lesen, 557
  - Sondervariablen schreiben, 559
  - Über Steuer-DB aktivieren und deaktivieren, 587
  - Variablen lesen, 554
  - Variablen schreiben, 555
  - Zugriff über PC, 571
- Benutzerdefinierte Webseiten aktualisieren, 553
- Benutzerdefinierte Webseiten anlegen, 552
- Beobachten
  - Beobachtungstabelle, 727, 729
  - Forcefunktion, 733
  - Forcetabelle, 732
  - KOP-Zustand, 727, 728
  - Speicherauslastung, 723
  - Startwerte eines DBs zurücksetzen, 728
  - Werte eines DBs erfassen, 728
  - Zykluszeit, 723
- Beobachtungstabelle
  - Ausgänge im Betriebszustand STOP freischalten, 732
  - Beobachten, 727
  - Forcen, 182
  - Funktionsweise, 729
  - Memory Card, 117
  - Werte auslösen, 731
- Bereich kopieren (MOVE\_BLK), 218
- Bereich ununterbrechbar kopieren (UMOVE\_BLK), 218
- Bestellnummern
  - CPUs, 883
  - CSM 1277 Ethernet Switch, 886
  - Dokumentation, 888
  - Erweiterungskabel, 887
  - Grundlegende HMI-Panels, 886
  - Klemmenblöcke, 887
  - Kommunikationsschnittstellen (CM, CB und CP), 885
  - Memory Cards, 886
  - PM 1207 Stromversorgung, 886
  - Programmiersoftware, 887
  - Signalboards (SB), 884
  - Signalmodule (SM), 883
  - Simulatoren, 887
  - Steckverbinder und Klemmenanschlüsse, 885
  - STEP 7, 887
  - Visualisierungssoftware, 887
  - WinCC, 887
- Betriebsstundenzähler, 265
- Betriebszustand, 40
  - Betriebszustände der CPU, 75
  - STOP/RUN wechseln, 723

- Betriebszustand RUN, 75, 79, 723
  - Bedienpanel, 40
  - Forcefunktion, 733
  - Schaltflächen in der Funktionsleiste, 40
- Betriebszustand STOP, 75, 723
  - Ausgänge im Betriebszustand STOP freischalten, 732
  - Bedienpanel, 40
  - Forcefunktion, 733
  - Schaltflächen in der Funktionsleiste, 40
- Bewegungssteuerung
  - Achse konfigurieren, 391
  - Hardware- und Software-Endschalter, 421
  - Konfigurationsparameter für die Referenzpunktfahrt, 427
  - MC\_ChangeDynamic, 418
  - MC\_CommandTable, 415
  - MC\_Halt, 404
  - MC\_Home, 401
  - MC\_MoveAbsolute, 406
  - MC\_MoveJog, 413
  - MC\_MoveRelative, 408
  - MC\_MoveVelocity, 410
  - MC\_Power, 397
  - MC\_Reset, 400
  - Referenzpunktfahrt (Abfolge bei der aktiven Referenzpunktfahrt), 429
  - Referenzpunktfahrt der Achse, 425
  - Übersicht, 387
- Bibliothek für das USS-Protokoll
  - Allgemeine Informationen zur Antriebseinrichtung, 656
  - Statuscodes, 654
  - Übersicht, 642
  - USS\_DRV, 647
  - USS\_PORT, 650
  - USS\_RPM, 651
  - USS\_WPM, 652
  - Voraussetzungen für den Einsatz, 644
- Bitverknüpfung, 185
- Boolesche Werte oder Bitwerte, 97
- Busstecker, 25
  
- C**
- CALCULATE, 35, 209
  - Analogwerte skalieren, 36
- CAN\_DINT (Verzögerungsalarm stornieren), 312
- CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 230
- CE-Zulassung, 743
- Char (Datentyp Character), 107
  
- CM 1241
  - Technische Daten RS232, 873
  - Technische Daten RS422/RS485, 873
- Codebaustein
  - Alarmer, 21, 751, 761, 771, 782
  - Anfangswert eines FB, 161
  - Anzahl der Codebausteine, 21, 73, 751, 761, 771, 782
  - Anzahl der OBs, 21, 751, 761, 771, 782
  - Bausteinaufrufe, 73
  - Beobachten, 751, 761, 771, 782
  - Codebausteine im Anwenderprogramm aufrufen, 158
  - DB (Datenbaustein), 73, 162
  - FB (Funktionsbaustein), 73, 161
  - FC (Funktion), 73, 160
  - Größe des Anwenderprogramms, 21, 751, 761, 771, 782
  - Gültige FC-, FB- und DB-Nummern, 73
  - Instanz-Datenbaustein (DB), 161
  - Knowhow-Schutz, 177
  - Kopierschutz, 178
  - Lineare und strukturierte Programme, 156
  - Mit CPU oder Memory Card verknüpfen, 178
  - Organisationsbaustein (OB), 159
  - Organisationsbausteine (OBs), 21, 751, 761, 771, 782
  - Schachtelungstiefe, 21, 751, 761, 771, 782
  - Überwachung, 21
  - Zähler (Anzahl und Speicherbedarf), 21, 752, 762, 772, 782
  - Zeiten (Anzahl und Speicherbedarf), 21, 752, 762, 772, 782
- Codebausteine, 158
- Codebausteine im Anwenderprogramm aufrufen, 158
- Codebausteine vergleichen, 725
- CONCAT (Verkettung), 283
- CONV (Umwandeln), 225
- Cookie siemens\_automation\_language, 584
- Cookie, siemens\_automation\_language, 584
- Cookie-Einschränkungen, Standard-Webseiten, 593
- COS (Cosinus), 216
- CPU
  - 1211C, Schaltpläne, 759
  - 1211C, technische Daten, 750
  - 1212C, Schaltpläne, 769
  - 1212C, technische Daten, 760
  - 1214C, Schaltpläne, 779
  - 1214C, technische Daten, 770
  - 1215C, Schaltpläne, 791
  - 1215C, technische Daten, 780
  - Anlaufparameter, 119

- Anlaufverarbeitung, 78  
 Anzahl der Kommunikationsverbindungen, 451  
 ASi,  
 ASi-Adresse,  
 AS-i-Port, 511  
 Auf Werkseinstellungen zurücksetzen, 722  
 Ausgänge im Betriebszustand STOP freischalten, 732  
 Bausteine aus einer Online-CPU kopieren, 179  
 Bausteine vergleichen und synchronisieren, 725  
 Bedienpanel, 40  
 Bedienpanel (Online-CPU), 723  
 Beobachten, 727  
 Beobachtungstabelle, 729  
 Betriebsstundenzähler, 265  
 Betriebszustand RUN und STOP, 723  
 Betriebszustände, 75  
 Darstellung Analogeingang (Spannung), 818, 848  
 Eigenschaft Uhrzeitsynchronisation, 153  
 Einbau, 51, 52  
 Erdung, 69  
 Erweiterungskabel, 61  
 Ethernet-Port, 147  
 Forcen, 732, 733  
 Gerätekonfiguration, 129  
 HMI-Geräte, 28  
 HSC-Konfiguration, 366  
 Impulsausgänge, 330  
 Impulskanäle konfigurieren, 332  
 Induktive Lasten, 71  
 IP-Adresse, 147  
 Klemmenblock, 60  
 Knowhow-Schutz, 177  
 Kommunikation mit HMI konfigurieren, 489  
 Kommunikationsarten, 449  
 Kommunikationsboard (CB), 25  
 Kommunikationslast, 89  
 Laden in CPU, 179  
 Laden in Gerät, 151  
 Lampenlasten, 70  
 LED-Anzeigen, 717  
 Leere Übertragungskarte, 127  
 Leistungsbedarf, 879  
 Leistungsbilanz, 47  
 MAC- und IP-Adressen anzeigen, 151  
 MAC-Adresse, 147, 151  
 Mehrere konfigurieren, 491  
 Memory Card, 117, 875  
 Memory Card stecken, 118  
 Module hinzufügen, 132  
 Module konfigurieren, 136  
 Netzwerkverbindung, 137  
 Neues Gerät hinzufügen, 130  
 Nicht spezifizierte CPU, 131  
 OBs verarbeiten, 159  
 Online, 721, 727  
 Online gehen, 719  
 Parameter konfigurieren, 133  
 Passwortschutz, 176  
 PROFIBUS, 507  
 PROFIBUS-Adresse, 507  
 PROFIBUS-Port, 507  
 PROFINET, 147  
 PROFINET IO, 495  
 Programmausführung, 73  
 Programmkarte, 117, 122  
 Programmkarte anlegen, 122  
 RD\_LOC\_T (Lokalzeit lesen), 263  
 RD\_SYS\_T (Systemzeit lesen), 263  
 Richtlinien für Potentialtrennung, 68  
 Schaltflächen RUN/STOP, 40  
 Schrittantwortzeiten, 756, 766, 776, 786  
 Schutzstufen, 176  
 Signalboard (SB), 25  
 Startwerte eines DBs zurücksetzen, 728  
 Thermischer Bereich, 46, 50  
 Übersicht, 19  
 Übertragungskarte, 117, 119  
 Übertragungskarte anlegen, 119  
 Verdrahtungsrichtlinien, 67, 69  
 Vergleichstabelle, 20  
 Verlorenes Passwort, 127  
 Vorgehensweise bei verlorenem Passwort, 127  
 Werte eines DBs erfassen, 728  
 WR\_SYS\_T (Systemzeit schreiben), 263  
 Zugriffsschutz, 176  
 Zuweisen einer IP-Adresse zu einer Online-CPU, 145  
 Zykluszeitkonfiguration, 89  
 Zykluszeitüberwachung, 87  
 CPU-Eigenschaften, benutzerdefinierte Webseiten, 567  
 CPU-Eigenschaften, benutzerdefinierte Webseiten (mehrere Sprachen), 587  
 CPU-Kommunikation, 452  
 CTD (Rückwärtszählen), 201  
 C-Tick-Zulassung, 745  
 CTRL\_PWM, 328  
 CTS, 620  
 CTU (Vorwärtszählen), 201  
 CTUD (Vorwärts- und Rückwärtszählen), 201  
 cULus-Zulassung, 744

- D**
- Date
    - Datentyp Date, 105
    - DTL (Datentyp Date and Time long), 106
  - Datenbaustein
    - CONF\_DATA, 482
    - Einzelner FB mit Multiinstanz-DBs, 162
    - Globaler Datenbaustein, 96, 162
    - Gültige DB-Nummern, 73
    - Instanz-Datenbaustein, 96
    - Organisationsbaustein (OB), 159
    - Startwerte zurücksetzen, 728
    - Übersicht, 73, 162
    - Werte erfassen, 728
  - Datenbausteine für benutzerdefinierte Webseiten
    - Fragmente importieren, 564
  - Datenbausteinststeuerung, 353
  - Datenhandierungsbausteine (DHBs), 162
  - Datenprotokoll
    - Beispielprogramm, 349
    - DataLogClose (Datenprotokoll schließen), 340
    - DataLogCreate (Datenprotokoll erstellen), 335
    - DataLogNewFile (Datenprotokoll basierend auf vorhandenem Datenprotokoll erstellen), 342
    - DataLogOpen (Datenprotokoll öffnen), 338
    - DataLogWrite (Datenprotokoll schreiben), 341
    - Datenprotokolle anzeigen, 344
    - Datensatzstruktur, 334
    - Grenzwerte für die Datenprotokollgröße, 346
    - Überblick über Datenprotokolle, 333
  - Datentyp DTL
    - Anweisungen für die Systemuhr, 263
  - Datentypen, 102
    - Any (Pointer), 111
    - Arrays, 108
    - Bool, Byte, Wort und Doppelwort, 103
    - Char (Zeichen) und String (Zeichenfolge), 107
    - PLC-Datentypeditor, 109
    - Pointer (Pointer), 110
    - Real, LReal (Gleitpunkt), 104
    - Struc, 109
    - Time, Date, TOD (Uhrzeit), DTL (Date and Time long), 105
    - Übersicht über Pointer-Datentypen, 110
    - USInt, SInt, UInt, Int, UDIInt, Dint (Ganzzahl), 104
    - Variant (Pointer), 113
  - Datenübertragung, anstoßen, 610
  - Datum
    - SET\_TIMEZONE (Zeitzone setzen), 267
    - T\_ADD (Zeiten addieren), 262
    - T\_COMBINE (Zeiten verknüpfen), 263
    - T\_CONV (Zeiten umwandeln), 261
    - T\_DIFF (Zeitdifferenz), 262
    - T\_SUB (Zeiten subtrahieren), 262
  - DB (Datenbaustein), 73, 162
    - Gültige DB-Nummern, 73
    - Startwerte zurücksetzen, 728
    - Werte erfassen, 728
  - DB-Fragmente (benutzerdefinierte Webseiten)
    - Generieren, 569
  - DBs für benutzerdefinierte Webseiten erstellen, 569
  - DBs für benutzerdefinierte Webseiten generieren, 569
  - DC
    - Erdung, 69
    - Induktive Lasten, 71
    - Richtlinien für Potentialtrennung, 68
    - Verdrahtungsrichtlinien, 67, 69
  - DEC (Dekrementieren), 213
  - DECO (Decodieren), 254
  - DELETE (Teilzeichenkette löschen), 285
  - DETACH, 305
  - DeviceStates, 318
  - Diagnose
    - Alarm-OB, 498
    - Anweisung LED, 316
    - Beobachtungstabelle, 729
    - DeviceStates, 318
    - Diagnosepuffer, 724
    - GET\_DIAG, 321
    - LED-Anzeigen, 717
    - ModuleStates, 319
    - Speicherauslastung, 723
    - Statusanzeige, 93
    - Zustand, 498
    - Zykluszeit, 723
  - Diagnosedaten
    - Auslesen mit GET\_DIAG, 321
  - Diagnosefehler-OB, 81
  - Diagnosepuffer, 94, 724
  - Digitale E/A
    - Konfiguration, 136
    - Statusanzeigen, 718
  - Digitales Signalmodul (SM)
    - SM 1221, 792
    - SM 1222, 795, 797
    - SM 1223, 802
  - DIS\_AIRT (Alarmbearbeitung deaktivieren), 314
  - DIV (Dividieren), 211
  - Dokumentation, 4
  - DPNRM\_DG, 302
  - DPRD\_DAT, 300
  - DPWR\_DAT, 300
  - Drag & Drop zwischen Editoren, 39



**E****E/A**

- Adressierung, 101
- Analoge Statusanzeigen, 718
- Darstellung Analogausgang (Spannung), 819, 849
- Darstellung Analogausgang (Strom), 819, 849
- Darstellung Analogeingang (Spannung), 818, 848
- Darstellung Analogeingang (Strom), 818, 848
- Digitale Statusanzeigen, 718
- Forcefunktion, 733
- Forcen, 732
- In einer Beobachtungstabelle beobachten, 729
- Induktive Lasten, 71
- Schrittantwortzeiten (CPU), 756, 766, 776, 786
- Schrittantwortzeiten (SB), 847
- Schrittantwortzeiten des Signalmoduls (SM), 817
- Zugriffsfehler, PROFINET, 499
- Zustand in KOP beobachten, 728
- Eigenschaft Uhrzeitsynchronisation, 153
- Ein- und Ausgänge
  - Beobachten, 727
- Einbau
  - Abstand, 46
  - CPU, 52
  - Erdung, 69
  - Erweiterungskabel, 61
  - Induktive Lasten, 71
  - Klemmenblock, 60
  - Kommunikationsboard (CB), 55
  - Kommunikationsmodul (CM), 59
  - Kühlung, 46
  - Lampenlasten, 70
  - Luftströmung, 46
  - Montageabmessungen, 50
  - Richtlinien, 45
  - Richtlinien für Potentialtrennung, 68
  - Signalboard (SB), 55
  - Signalmodul (SM), 25, 57
  - Thermischer Bereich, 46, 50
  - TS-Adapter auf DIN-Schiene, 66
  - Übersicht, 45, 51
  - Verdrahtungsrichtlinien, 67, 69
  - Wandmontage des TS-Adapters, 66
- Einfügen eines Geräts
  - Nicht spezifizierte CPU, 131
- Einfügen von Anweisungen
  - Drag & Drop, 33
  - Drag & Drop zwischen Editoren, 39
  - Favoriten, 34
- Eingangssimulatoren, 876
- Eingeschränkte TSAPs und Portnummern, 487
- Einschaltverzögerung (TON), 193
- Funktionsweise, 197
- Einschränkungen
  - Benutzerdefinierte Webseiten, 571
  - Webserver, 591
- Einstellungen, 39
- Einstellungen von STEP 7 ändern, 39
- Elektromagnetische Verträglichkeit (EMV), 746
- EN und ENO (Signalfluss), 174
- EN\_AIRT (Alarmbearbeitung aktivieren), 314
- ENCO (Encodieren), 254
- Endbedingungen, 626
- Entwerfen einer Automatisierungslösung mit einem PLC-Gerät, 155, 156
- Enum-Typen definieren, benutzerdefinierte Webseiten, 561, 562
- Enum-Typen in benutzerdefinierten Webseiten, 561, 562
- Enum-Typen zuweisen, benutzerdefinierte Webseiten, 562
- Ereignisausführung, 81
- Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 37
- Erkennung zum Laden einer Online-CPU, 131
- Ersetzen von Modulen, 43
- Erweiterbare Anweisungen, 37
- Erweiterung der Leistungsmerkmale der S7-1200, 22
- Erweiterungskabel
  - Ausbau, 61
  - Einbau, 61
- Ethernet
  - Ad-hoc-Modus, 458
  - Anzahl der Kommunikationsverbindungen, 451
  - DPNRM\_DG, 302
  - DPRD\_DAT, 300
  - DPWR\_DAT, 300
  - GET, 518
  - IP-Adresse, 147
  - Kommunikationsarten, 449
  - MAC-Adresse, 147
  - Netzwerkverbindung, 137
  - PUT, 518
  - RALRM, 293
  - RDREC, 290
  - T\_CONFIG, 479
  - TCON, 466
  - TDISCON, 466
  - TRCV, 466
  - TRCV\_C, 459
  - TSEND, 466
  - TSEND\_C, 459
  - TURCV, 474
  - TUSEND, 474

Übersicht, 457  
Verbindungs-IDs, 454  
WRREC, 290  
Ethernet-Protokolle, 457  
Verbindungen mit mehreren Teilnehmern, 522  
EXP (Exponentialwert), 216  
EXPT (Potenzieren), 216

**F**

Fallende Flanke, 191  
FAQs, 4  
Favoriten, Funktionsleiste, 34  
FB (Funktionsbaustein)  
Gültige FB-Nummern, 73  
Übersicht, 73  
FC (Funktion), 73, 160  
Gültige FC-Nummern, 73  
Fehler  
Diagnosefehler, 86  
Zeitfehler, 84  
Fehler der Empfangskonfiguration, 610  
Fehler der Sendekonfiguration, 605  
Fehlerbehebung  
Diagnosepuffer, 724  
LED-Anzeigen, 717  
Fehlercodes  
Häufige Fehler bei erweiterten Anweisungen, 356  
RALRM, 296  
RDREC, 296  
WRREC, 296  
FeldLesen, 221  
FeldSchreiben, 221  
FILL\_BLK, 223  
FIND (Teilzeichenkette finden), 288  
Firmware-Update, 124  
FLOOR, 230  
Flusskontrolle, 620  
Konfiguration, 620  
FM-Zertifizierung, 744  
Forcen, 732, 733  
Beobachtungstabelle, 729  
Eingänge der Peripherie, 732, 733  
Eingänge und Ausgänge, 733  
Memory Card, 117  
Speicherbereich E, 732, 733  
Zyklus, 733  
Forcetabelle  
Eingänge der Peripherie ansprechen, 732  
Forcefunktion, 733  
Forcen, 732  
FRAC (Nachkommastellen), 216

Fragmente (benutzerdefinierte Webseiten)  
Mit AWP-Befehl importieren, 564  
Mit AWP-Befehlen erstellen, 563  
Frequenz, Taktbits, 94  
Funktion (FC)  
Codebausteine im Anwenderprogramm aufrufen, 158  
Gültige FC-Nummern, 73  
Knowhow-Schutz, 177  
Lineare und strukturierte Programme, 156  
Übersicht, 73, 160  
Funktionsbaustein (FB)  
Anfangswert, 161  
Ausgangsparameter, 161  
Codebausteine im Anwenderprogramm aufrufen, 158  
Einzelner FB mit Multiinstanz-DBs, 162  
Gültige FB-Nummern, 73  
Instanz-Datenbaustein, 161  
Knowhow-Schutz, 177  
Lineare und strukturierte Programme, 156  
Übersicht, 73, 161  
FUP (Funktionsplan), 166

**G**

Gerät  
PROFINET IO, 495  
Gerätekonfiguration, 129, 488  
ASi,  
AS-i-Port, 512  
CPU konfigurieren, 133  
Eigenschaft Uhrzeitsynchronisation (PROFINET), 153  
Erkennen, 131  
Ethernet-Port, 147  
Laden in CPU, 179  
Module hinzufügen, 132  
Module konfigurieren, 136  
Netzwerkverbindung, 137  
Neues Gerät hinzufügen, 130  
Nicht gesteckte Module, 43  
PROFIBUS, 507  
PROFIBUS-Port, 507  
PROFINET, 147  
Gerätenamen  
PROFINET IO, 496  
GET, 518  
Verbindung konfigurieren, 139  
GET\_DIAG, 321  
GetError, 249  
GetErrorID, 250

Gleitpunktarithmetik, 216  
 Globale Bibliothek  
   Überblick über das USS-Protokoll, 642  
 Globaler Datenbaustein, 96, 162  
 Grenzwert, 216  
 Grundlegende Panels (HMI), 28

**H**

Handbücher, 4  
 Hardware-Flusskontrolle, 620  
 Hardwarekonfiguration, 129  
   ASi,  
   AS-i-Port, 512  
   CPU konfigurieren, 133  
   Erkennen, 131  
   Ethernet-Port, 147  
   Laden in CPU, 179  
   Module hinzufügen, 132  
   Module konfigurieren, 136  
   Netzwerkverbindung, 137  
   Neues Gerät hinzufügen, 130  
   PROFIBUS, 507  
   PROFIBUS-Port, 507  
   PROFINET, 147  
 Herunterladen  
   Firmware-Update, 124  
   Siemens-Sicherheitszertifikat auf PC, 537, 593  
 HMI-Geräte  
   Netzwerkverbindung, 137  
   PROFINET-Kommunikation konfigurieren, 489  
   Übersicht, 28  
 Hotline, 3  
 HSC (schneller Zähler)  
   Funktionsweise, 357, 359  
   Konfiguration, 366  
 HTA (Hexadezimalzahl in ASCII-Zeichenkette wandeln), 281  
 HTML-Liste, Beispiel für benutzerdefinierte Webseiten, 578  
 HTML-Seiten, benutzerdefiniert, 551  
   Aktualisieren, 553  
   Entwickeln, 552  
   Seitenanordnung, 567  
   Speicherorte für Sprachen, 587  
   Zugriff auf S7-1200 Daten, 553  
 HTTP-Variablen lesen, 557  
 HTTP-Verbindungen, Webserver, 591

**I**

Impuls (TP), 193  
   Funktionsweise, 197  
 Impulsausgänge, 330  
 IN\_RANGE (innerhalb eines Bereichs), 208  
 INC (Inkrementieren), 213  
 Induktive Lasten, 71  
 Informationsquellen, 4  
 INSERT (Teilzeichenkette einfügen), 286  
 Installation  
   Leistungsbilanz, 47  
   TS-Adapter SIM-Karte, 64  
   TS-Adapter und TS-Modul, 63  
 Instanz-Datenbaustein, 96  
 Invertieren (INV), 253  
 IP-Adresse, 147, 148  
   CPU online konfigurieren, 721  
   Gerätekonfiguration, 133  
   Konfigurieren, 147  
   MAC-Adresse, 147  
   Online zuweisen, 145  
   Zuordnen, 143, 150  
 IP-Adresse des Routers, 148  
 IP-Router, 147  
 ISO on TCP  
   Ad-hoc-Modus, 458  
 ISO-on-TCP  
   Parameter, 140  
   Verbindungs-IDs, 454  
   Verbindungskonfiguration, 138  
 ISO-on-TCP-Protokoll, 457

**J**

JavaScript-Einschränkungen, Standard-Webseiten, 592  
 JMP, 242  
 JMP\_LIST, 243  
 JMPN, 242

**K**

Kalender, 261  
 Kein Neustart, 75  
 Klemmenblock, 60  
 Knowhow-Schutz  
   Passwortschutz, 177  
 Kommunikation  
   Abfragearchitektur, 632  
   Aktiv/passiv, 138, 140, 523  
   Anzahl der Verbindungen (PROFINET), 451

- ASi-Adresse,
  - Eigenschaft Uhrzeitsynchronisation (PROFINET), 153
  - Flusskontrolle, 620
  - Hardwareanschluss, 487
  - IP-Adresse, 147
  - Kommunikationslast, 89
  - Konfiguration, 138, 140, 523
  - MAC-Adresse, 147
  - Netzwerk, 487
  - Netzwerkverbindung, 137
  - Parameter, 140, 543
  - PROFIBUS-Adresse, 507
  - PROFINET und PROFIBUS, 449
  - Sende- und Empfangsparameter, 622
  - Statistik, 543
  - TCON\_Param, 140
  - Verbindungs-IDs, 454
  - Zykluszeit, 89
- Kommunikationsboard (CB)
  - Ausbau, 55
  - CB 1241 RS485, 870
  - Einbau, 55
  - Gerätekonfiguration, 129
  - Konfiguration von Parametern, 136
  - LED-Anzeigen, 597, 717
  - Module hinzufügen, 132
  - Programmieren, 631
  - RS485, 597
  - Übersicht, 25
  - Vergleichstabelle, 22
- Kommunikationsmodul
  - CM 1241 RS422/RS485, Technische Daten, 873
- Kommunikationsmodul (CM)
  - AS-i-Mastermodul CM1243-2 hinzufügen, 510
  - Ausbau, 59
  - CM 1241 RS232, technische Daten, 873
  - CM 1243-5 (DP-Master) hinzufügen, 506
  - Datenempfang, 614
  - Einbau, 59
  - Gerätekonfiguration, 129
  - Konfiguration des PtP-Beispielprogramms, 634
  - Konfiguration von Parametern, 136
  - LED-Anzeigen, 597, 717
  - Leistungsbedarf, 879
  - Module hinzufügen, 132
  - Programmieren, 631
  - RS232 und RS485, 597
  - Übersicht, 26
  - Vergleichstabelle, 22
- Kommunikationsprozessor (CP)
  - Gerätekonfiguration, 129
- Konfiguration von Parametern, 136
  - Module hinzufügen, 132
  - Übersicht, 26
  - Vergleichstabelle, 22
- Kommunikationsschnittstellen
  - CB 1241 RS485, 870
  - CM 1241 RS232, 873
  - Gerätekonfiguration, 129
  - Konfiguration, 618
  - LED-Anzeigen, 717
  - Module hinzufügen, 132
  - Programmieren, 631
  - RS232 und RS485, 597
  - Vergleichstabelle der Module, 22
- Kompensation der kalten Verbindungsstelle
  - Thermoelement, 824, 852
- Konfiguration
  - Anlaufparameter, 119
  - ASi,
    - ASi-Adresse,
      - AS-i-Port, 511
    - Benutzerdefinierte Webseiten, 567
    - Benutzerdefinierte Webseiten (mehrere Sprachen), 587
    - CPU-Parameter, 133
    - Eigenschaft Uhrzeitsynchronisation (PROFINET), 153
    - Erkennen, 131
    - Ethernet-Port, 147
    - HSC (schneller Zähler), 366
    - IP-Adresse, 147
    - Kommunikation PLC/PLC, 491
    - Kommunikationslast, 89
    - Kommunikationsschnittstellen, 618
    - Laden in CPU, 179
    - MAC-Adresse, 147
    - Meldung empfangen, 623
    - Module, 136
    - Module hinzufügen, 132
    - Netzwerkverbindung, 137
    - PROFIBUS, 507
    - PROFIBUS-Adresse, 507
    - PROFIBUS-Port, 507
    - PROFINET, 147
    - Schnittstellen, 618
    - Zykluszeit, 87
  - Konfiguration der gesendeten Meldung, 622
  - Konfiguration Empfangsparameter, 495
  - Konfiguration für Meldungssendung, 622
  - Konfiguration Sendeparameter, 138, 494, 523
  - Konsistenzprüfung, 183
  - Kontaktdaten, 3

- Konventionen bei Anführungszeichen, Webserver, 565
- KOP (Kontaktplan)
- Beobachten, 727, 728
  - Programmiereditor, 728
  - Übersicht, 165
  - Zustand, 727, 728, 732
- Kopierschutz
- Mit CPU oder Memory Card verknüpfen, 178
- Kühlen, 46
- Kunden-Support, 3
- L**
- Laden aus CPU
- Anwenderprogramm, 179
  - Bausteine aus einer Online-CPU kopieren, 179
- Laden in CPU
- Anwenderprogramm, 179
  - DBs für benutzerdefinierte Webseiten, 571
  - MAC- und IP-Adressen anzeigen, 151
  - Projekt, 179
- Ladespeicher, 20
- CPU 1211C, 750
  - CPU 1212C, 760
  - CPU 1214C, 770
  - CPU 1215C, 780
  - Memory Card, 117
  - Programmkarte, 117
  - Übertragungskarte, 117
- Ladespeicher, benutzerdefinierte Webseiten, 571
- Lampenlasten, 70
- Länge
- Meldung, 629
- Länge m, 629
- Länge n, 629
- Latenz, 81
- Laufzeitfehler beim Senden, 612
- Lebensdauer eines Relais, 749
- LED (LED-Status lesen), 316
- LED-Anzeigen
- Anweisung LED, 316
  - CPU-Status, 717
  - Kommunikationsschnittstelle, 597, 717
- Leerzeile, 623
- LEFT (Linke Teilzeichenkette), 284
- Leistungsbedarf
- Berechnen der Leistungsbilanz, 881
  - Formular für Berechnungen, 882
  - Leistungsbilanz, 879
- Leistungsbilanz, 47
- Beispiel, 881
  - Formular für Berechnungen, 882
- Übersicht, 879
- Leitung im Leerlauf, 624
- LEN (Länge), 282
- Lineare Programmierung, 156
- LN (Natürlicher Logarithmus), 216
- Luftströmung, 46
- M**
- MAC- und IP-Adressen anzeigen, 151
- MAC-Adresse, 147, 151
- Manuelle Steuerung von DB-Fragmenten, 587
- MAX (Maximum), 215
- Maximale Anzahl Webserver-Verbindungen, 591
- Maximale Meldungslänge, 628
- MB\_CLIENT, 662
- MB\_COMM\_LOAD, 679
- MB\_MASTER, 682
- MB\_SERVER, 669
- MB\_SLAVE, 689
- MC\_ChangeDynamic, 418
- MC\_CommandTable, 415
- MC\_Halt, 404
- MC\_Home, 401
- MC\_MoveAbsolute, 406
- MC\_MoveJog, 413
- MC\_MoveRelative, 408
- MC\_MoveVelocity, 410
- MC\_Power, 397
- MC\_Reset, 400
- Mehrere AWP-Variablendefinitionen, 565
- Meldungsbeginn, 624
- Meldungsempfang konfigurieren, 623
- PtP-Beispielprogramm, 635
- Meldungsende, 626
- Meldungskonfiguration
- Anweisungen, 631
  - Empfangen, 623
  - Senden, 622
- Meldungslänge, 628
- Meldungssendung konfigurieren, 622
- PtP-Beispielprogramm, 634
- Memory Card
- Firmware-Update, 124
  - Funktionsweise, 117
  - In CPU einfügen, 118
  - Leere Übertragungskarte für ein verlorenes Passwort, 127
  - Programmkarte, 122
  - Übersicht, 117
  - Übertragungskarte, 119
- Memory Card in CPU stecken, 118

MID (Mittlere Teilzeichenkette), 284  
MIN (Minimum), 215  
Mit CPU oder Memory Card verknüpfen, 178  
MOD (Modulo), 212  
MODBUS  
    Beispiel für Modbus-Slave, 697  
    MB\_CLIENT, 662  
    MB\_COMM\_LOAD, 679  
    MB\_MASTER, 682  
    MB\_SERVER, 669  
    MB\_SLAVE, 689  
    Versionen, 38, 678  
Module  
    CB 1241 RS485, 870  
    CM 1241 RS232, 873  
    Kommunikationsboard (CB), 25  
    Kommunikationsmodul (CM), 26  
    Kommunikationsprozessor (CP), 26  
    Parameter konfigurieren, 136  
    SB 1221 DI 4, 200 kHz, 832  
    SB 1222 DO 4, 200 kHz, 200 kHz, 834  
    SB 1223 DI 2 / DO 2, 840  
    SB 1223 DI 2 / DO 2, 200 kHz, 837  
    SB 1231 AI 1 x 12 Bit, 843  
    SB 1231 AI 1 x 16 Bit RTD, 854  
    SB 1231 AI 1 x 16 Bit Thermoelement-  
    Signalboard, 850  
    SB 1232 AO 1x12 Bit, 845  
    Signalboard (SB), 25  
    Signalmodul (SM), 25  
    Signalmodul SM 1231 AI 8 x 16 Bit TC, 820  
    SM 1221, 792  
    SM 1222, 795, 797  
    SM 1222 DO8 RLS, Umschaltung, 795  
    SM 1223, 802  
    SM 1231 AI 4 x 16 Bit TC, 820  
    SM 1231 AI 4 x RTD x 16 Bit, 826  
    SM 1231 AI 8 x 13 Bit, 808  
    SM 1231 AI 8 x RTD x 16 Bit, 826  
    SM 1232 AO 2 x 14 Bit, 812  
    SM 1232 AO 4 x 14 Bit, 812  
    SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 814  
    Technische Daten der CPU 1211C, 750  
    Technische Daten der CPU 1212C, 760  
    Technische Daten der CPU 1214C, 770  
    Technische Daten der CPU 1215C, 780  
    Thermischer Bereich, 46, 50  
    Vergleichstabelle, 22  
ModuleStates, 319  
Montage  
    Abmessungen, 50  
    Abstand, 46

CPU, 52  
Erdung, 69  
Erweiterungskabel, 61  
Induktive Lasten, 71  
Klemmenblock, 60  
Kommunikationsboard (CB), 55  
Kommunikationsmodul (CM), 59  
Kühlung, 46  
Lampenlasten, 70  
Luftströmung, 46  
Richtlinien, 45  
Signalboard (SB), 55  
Signalmodul (SM), 57  
Thermischer Bereich, 46, 50  
Trennung, 68  
Übersicht, 51  
Verdrahtungsrichtlinien, 67, 69  
MOVE, 218  
MRES  
    Bedienpanel, 40  
MUL (Multiplizieren), 211  
MUX (Multiplexen), 256  
My Documentation Manager, 4

## N

N\_TRIG, 192  
NEG (Negation), 213  
Network Time Protocol (NTP), 152  
Netzwerkcommunication, 487  
Netzwerkverbindung  
    Konfiguration, 137  
    Mehrere CPUs, 491, 492, 496, 506, 511  
Netzwerkverbindung erstellen, 137  
Neues Gerät hinzufügen  
    CPU, 130  
    Nicht spezifizierte CPU, 131  
    Vorhandene Hardware erkennen, 131  
Nicht gesteckte Module, 43  
Nicht spezifizierte CPU, 131  
NORM\_X (Normalisieren), 231

## O

Online  
    Bedienpanel, 40, 723  
    Beobachten, 727  
    Beobachtungstabelle, 727, 728, 729  
    Diagnosepuffer, 724  
    Forcefunktion, 733  
    Forcen, 732

- IP-Adresse, 721
- IP-Adresse zuordnen, 145
- Online gehen, 719
- Schaltflächen RUN/STOP, 40
- Speicherauslastung, 723
- Startwerte eines DBs zurücksetzen, 728
- Tageszeit, 721
- Tools, 726
- Vergleichen und synchronisieren, 725
- Werte eines DBs erfassen, 728
- Zustand, 727, 728
- Zykluszeit, 723
- Online- und Diagnose-Tools
  - Laden im Betriebszustand RUN, 735
- Online- und Offline-CPU's vergleichen und synchronisieren, 725
- Online-Gerätenamen
  - PROFINET IO, 720
- OPC, Konfiguration, 705
- OR, 252
- Ordner, Sprachen für benutzerdefinierte Webseiten, 584
- Organisationsbaustein
  - Anlaufverarbeitung, 78
  - Aufrufen, 80
  - Bearbeiten, 159
  - Codebausteine im Anwenderprogramm aufrufen, 158
  - Erstellen, 160
  - Funktion, 80
  - Funktionsweise konfigurieren, 160
  - Knowhow-Schutz, 177
  - Lineare und strukturierte Programmierung, 156
  - Mehrere Zyklus-OBs, 160
  - Prioritätsklassen, 80
  - Übersicht, 73
- Ortszeit
  - RD\_LOC\_T (Lokalzeit lesen), 263
- OUT\_RANGE (außerhalb eines Bereichs), 208
  
- P**
- P\_TRIG, 192
- Panels (HMI), 28
- Parameter BUFFER, SEND\_PTP, 612
- Parameter LENGTH, SEND\_PTP, 612
- Parameterkonfiguration
  - Empfangen, 495
  - LENGH und BUFFER für SEND\_PTP, 612
  - Senden, 494
- Parametrieren, 161
- Parität, 620
- Passive/aktive Kommunikation
  - Parameter, 140
  - Partner konfigurieren, 138, 523
  - Verbindungs-IDs, 454
- Passwortschutz
  - Codebaustein, 177
  - CPU, 176
  - Kopierschutz, 178
  - Leere Übertragungskarte, 127
  - Mit CPU oder Memory Card verknüpfen, 178
  - Verlorenes Passwort, 127
  - Zugriff auf die CPU, 176
- Pause, 624
- Peripheriezugriffsalarme, 499
- PID
  - Algorithmus PID\_3Step, 368, 376
  - Algorithmus PID\_Compact, 368, 370
  - PID\_3STEP, 376
  - PID\_Compact, 370
  - Übersicht, 367
- PLC
  - Anlaufverarbeitung, 78
  - Bausteine aus einer Online-CPU kopieren, 179
  - Beobachten, 727
  - Beobachtungstabelle, 729
  - Betriebsstundenzähler, 265
  - Betriebszustände, 75
  - CPU 1211C, 750
  - CPU 1212C, 760
  - CPU 1214C, 770
  - CPU 1215C, 780
  - Eigenschaft Uhrzeitsynchronisation, 153
  - Einbau, 51, 52
  - Entwerfen einer Automatisierungslösung, 155
  - Erweiterungskabel, 61
  - Forcefunktion, 733
  - Forcen, 732
  - Gerätekonfiguration, 129
  - HSC-Konfiguration, 366
  - Klemmenblock, 60
  - Knowhow-Schutz, 177
  - Kommunikationslast, 89
  - Laden in CPU, 179
  - Leistungsbilanz, 47
  - Memory Card, 117, 875
  - Module hinzufügen, 132
  - RD\_LOC\_T (Lokalzeit lesen), 263
  - RD\_SYS\_T (Systemzeit lesen), 263
  - Übersicht über die CPU, 19
  - Variablen, 96
  - Vergleichen und synchronisieren, 725
  - WR\_SYS\_T (Systemzeit schreiben), 263

- Zuweisen einer IP-Adresse zu einer Online-CPU, 145
- Zykluszeit, 89
- Zykluszeit, 89
- PLC-Gerät
  - Bausteine verwenden, 156
- Podcasts, 4
- Pointer
  - Datentyp Pointer, 110
  - Datentyp Variant, 113
  - Datenyp Any, 111
  - Übersicht über Pointer, 110
- PORT\_CFG (Portkonfiguration), 602
- Portalansicht, 32
- Portkonfiguration, 618
  - Error, 603
  - PtP-Beispielprogramm, 634
- Portnummer, 457
- Portnummern
  - eingeschränkt, 487
- Priorität
  - Priorität bei der Verarbeitung, 81
  - Prioritätsklassen, 80
- Prioritätsklasse, 80
- PROFIBUS
  - Anweisungen für die dezentrale Peripherie, 289
  - CM 1242-5 (DP-Slave), 500
  - CM 1243-5 (DP-Master), 500
  - CM 1243-5 (DP-Master) hinzufügen, 506
  - DPNRM\_DG, 302
  - DPRD\_DAT, 300
  - DP-Slave hinzufügen, 506
  - DPWR\_DAT, 300
  - Eigenschaften PROFIBUS-Adresse, 508
  - GET, 518
  - Master, 500
  - Netzwerkverbindung, 137, 506
  - PROFIBUS-Adresse, 507
  - PUT, 518
  - RALRM, 293
  - RDREC, 290
  - S7-Verbindung, 522
  - Slave, 500
  - WRREC, 290
- PROFIBUS-Adresse, 507, 508
  - Konfigurieren, 507
- PROFINET
  - Ad-hoc-Modus, 458
  - Anweisungen für die dezentrale Peripherie, 289
  - Anzahl der Kommunikationsverbindungen, 451
  - CPU-zu-CPU-Kommunikation, 491
  - Diagnose, 499
  - DPRD\_DAT, 300
  - DPWR\_DAT, 300
  - Eigenschaft Uhrzeitsynchronisation, 153
  - Eigenschaften der Ethernet-Adresse, 148
  - Gerätenamen und Adressierung, 154
  - GET, 518
  - IP-Adresse, 147
  - IP-Adresse konfigurieren, 133
  - Kommunikation PLC/PLC, 491
  - Kommunikation zwischen CPU und HMI-Gerät konfigurieren, 489
  - Kommunikationsarten, 449
  - MAC-Adresse, 147
  - Netzwerkverbindung, 137, 491, 492, 496
  - PUT, 518
  - RALRM, 293
  - RDREC, 290
  - S7-Verbindung, 522
  - Startzeit des Systems, 153
  - T\_CONFIG, 479
  - TCON, 466
  - TDISCON, 466
  - Testen eines Netzwerks, 150
  - TRCV, 466
  - TRCV\_C, 459
  - TSEND, 466
  - TSEND\_C, 459
  - TURCV, 474
  - TUSEND, 474
  - Übersicht, 457
  - Uhrzeitsynchronisierung, 133
  - Verbindungs-IDs, 454
  - WRREC, 290
  - Zuweisung der IP-Adresse, 154
- PROFINET IO
  - CPU zuweisen, 496
  - Geräte, 495
  - Gerätenamen, 496
  - Gerätenamen online zuweisen, 720
  - Gerätenamen zuweisen, 496
  - Hinzufügen eines Geräts, 495
  - Online-Gerätenamen, 720
- PROFINET RT, 457
- Programm
  - Bausteine aus einer Online-CPU kopieren, 179
  - Codebausteine im Anwenderprogramm aufrufen, 158
  - Laden in CPU, 179
  - Lineare und strukturierte Programme, 156
  - Memory Card, 117
  - Mit CPU oder Memory Card verknüpfen, 178
  - Organisationsbaustein (OB), 159



- Passwortschutz, 177
- Prioritätsklassen, 80
- Startwerte eines DBs zurücksetzen, 728
- Werte eines DBs erfassen, 728
- Programm beobachten, 182
- Programm testen, 182
- Programmausführung, 73
- Programmiereditor
  - Beobachten, 728
  - Startwerte eines DBs zurücksetzen, 728
  - Werte eines DBs erfassen, 728
  - Zustand, 728
- Programmierung
  - Algorithmus PID\_3Step, 368, 376
  - Algorithmus PID\_Compact, 368, 370
  - Anfangswert eines FB, 161
  - Anzahl der Codebausteine, 73
  - Bausteinaufrufe, 73
  - Betriebsstundenzähler, 265
  - Betriebszustände der CPU, 75
  - Codebausteine im Anwenderprogramm aufrufen, 158
  - Codebausteine vergleichen und synchronisieren, 725
  - Datenbaustein (DB), 73
  - Drag & Drop zwischen Editoren, 39
  - Einfügen von Anweisungen, 33
  - Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 37
  - Erweiterbare Anweisungen, 37
  - Favoriten, 34
  - Funktion (FC), 160
  - Funktionsbaustein (FB), 73, 161
  - FUP (Funktionsplan), 166
  - Gültige FC-, FB- und DB-Nummern, 73
  - Instanz-Datenbaustein (DB), 161
  - KOP (Kontaktplan), 165
  - Lineares Programm, 156
  - Mit CPU oder Memory Card verknüpfen, 178
  - Nicht gesteckte Module, 43
  - Nicht spezifizierte CPU, 131
  - Organisationsbaustein (OB), 159
  - PID\_3STEP, 376
  - PID\_Compact, 370
  - Prioritätsklassen, 80
  - PtP-Anweisungen, 631
  - RD\_LOC\_T (Lokalzeit lesen), 263
  - RD\_SYS\_T (Systemzeit lesen), 263
  - SCL (Structured Control Language), 166, 167, 168
  - Signalfluss (EN und ENO), 174
  - Strukturiertes Programm, 156
  - Übersicht über den PID-Regler, 367
  - WR\_SYS\_T (Systemzeit schreiben), 263
  - Zähler, 201
- Programminformationen
  - In der Aufrufstruktur, 183
- Programmkarte
  - Anlaufparameter konfigurieren, 119
  - Anlegen, 122
  - Bestellnummer, 875
  - Funktionsweise, 117
  - In CPU einfügen, 118
  - Übersicht, 117
- Programmsteuerung (SCL), 234
  - CASE, 236
  - CONTINUE, 240
  - EXIT, 241
  - FOR, 237
  - GO TO, 241
  - IF-THEN, 235
  - REPEAT, 239
  - RETURN, 242
  - WHILE, 238
- Programmstruktur, 158
  - Codebausteine aufrufen, 158
- Projekt
  - Codebaustein schützen, 177
  - Laden in CPU, 179
  - Leere Übertragungskarte, 127
  - Mit CPU oder Memory Card verknüpfen, 178
  - Programmkarte, 122
  - Übertragungskarte, 119
  - Vergleichen und synchronisieren, 725
  - Verlorenes Passwort, 127
  - Zugang zur CPU einschränken, 176
  - Zugriffsschutz, 176
- Projektansicht, 32, 33
- Protokoll
  - Frei programmierbare Kommunikation, 599
  - ISO on TCP, 457
  - Kommunikation, 599
  - Modbus, 599
  - PROFINET RT, 457
  - TCP, 457
  - UDP, 457
  - USS, 599
- Protokoll der frei programmierbaren Kommunikation, 599
- Prozessabbild
  - Beobachten, 727, 728
  - Forcefunktion, 733
  - Forcen, 732
  - Zustand, 727, 728, 732
- Prozessalarm-OB, 80

- PTO (Impulsfolge)
  - CTRL\_PWM, 328
  - Funktionsweise, 330
  - Impulskanäle konfigurieren, 332
  - Kann nicht geforct werden, 734
- PtP-Anweisungen Rückgabewerte, 600
- PtP-Fehlerklassen, 602
- PtP-Kommunikation, 599
  - Anschlüsse konfigurieren, 618
  - Beispielprogramm, 633
  - Beispielprogramm, Ausführen, 642
  - Beispielprogramm, STEP 7-Programmierung, 640
  - Konfiguration des Beispielprogramms, 634
  - Parameter konfigurieren, 622
  - Programmieren, 631
  - Terminalemulator für Beispielprogramm, 641
- Punkt-zu-Punkt-Kommunikation, 599
- Punkt-zu-Punkt-Programmierung, 631
- PUT, 518
  - Verbindung konfigurieren, 139
- PWM (Impulsdauermodulation)
  - CTRL\_PWM, 328
  - Funktionsweise, 330
  - Impulskanäle konfigurieren, 332
  - Kann nicht geforct werden, 734
- Q**
- QRY\_CINT (Weckalarm abfragen), 310
- Querverweise, 183
  - Einleitung, 183
  - Verwendung, 183
- R**
- RALRM, 293, 296
- RCV\_CFG (Empfangskonfiguration), 605
- RCV\_PTP (Punkt-zu-Punkt-Daten empfangen), 614
- RCV\_RST (Empfänger zurücksetzen), 615
- RD\_LOC\_T (Lokalzeit lesen), 263
- RD\_SYS\_T (Systemzeit lesen), 263
- RDREC, 290, 296
- RE\_TRIGR, 247
- READ\_DBL, 353
- Remanenter Speicher, 20, 90
  - CPU 1211C, 750
  - CPU 1212C, 760
  - CPU 1214C, 770
  - CPU 1215C, 780
- REPLACE (Teilzeichenkette ersetzen), 287
- Richtlinien
  - Einbau der CPU, 52
  - Erdung, 69
  - Induktive Lasten, 71
  - Installation, 45
  - Lampenlasten, 70
  - Trennung, 68
  - Verdrahtungsrichtlinien, 67, 69
  - Vorgehensweise für den Einbau, 51
- Richtlinien für Potentialtrennung, 68
- RIGHT (Rechte Teilzeichenkette), 284
- ROL und ROR (Links rotieren und Rechts rotieren), 259
- ROUND, 229
- RS232- und RS485-Kommunikationsmodule, 597
- RT (Zeit zurücksetzen), 193
- RTS, 620
- RTS immer ein, 621
- RTS-Ausschaltverzögerung, 623
- RTS-Einschaltverzögerung, 623
- RTS-geschaltet, 620
- Rückgabewert (RET), 246
- Rückgabewerte
  - Anweisungen für die offene Benutzerkommunikation, 486
  - PtP-Anweisungen, 600
- Rückgabewerte der Anweisungen für die offene Benutzerkommunikation, 486
- Rückgabewerte Empfangs-Laufzeit, 614
- Rücksetzen, 188
- S**
- S\_CONV (Zeichenkette konvertieren), 269
- S\_MOV (Zeichenkette verschieben), 269
- S7-1200
  - Abstand, 46
  - Anlaufparameter, 119
  - Anlaufverarbeitung, 78
  - ASi,
    - ASi-Adresse,
    - AS-i-Port, 511
  - Bedienpanel, 40
  - Beobachten, 727
  - Betriebszustände, 75
  - Codebausteine vergleichen, 725
  - CPU-Parameter konfigurieren, 133
  - Einbau, 51
  - Einbau der CPU, 52
  - Einbau eines CBs, 55
  - Einbau eines CM, 59
  - Einbau eines SB, 55
  - Einbau eines SM, 57

- Erdung, 69
- Erweiterungskabel, 61
- Ethernet-Port, 147
- Forcefunktion, 733
- Forcen, 732
- Funktionsweise, 729
- Gerätekonfiguration, 129
- HMI-Geräte, 28
- HSC-Konfiguration, 366
- Impulsausgänge, 330
- Induktive Lasten, 71
- IP-Adresse, 147
- Klemmenblock, 60
- Knowhow-Schutz, 177
- Kommunikationsboard (CB), 25
- Kommunikationslast, 89
- Kommunikationsmodul (CM), 26
- Kommunikationsprozessor (CP), 26
- Kühlung, 46
- Lampenlasten, 70
- Leere Übertragungskarte für ein verlorenes Passwort, 127
- Leistungsbilanz, 47
- Luftströmung, 46
- MAC-Adresse, 147
- Memory Card, 875
- Module, 22
- Module hinzufügen, 132
- Module konfigurieren, 136
- Montageabmessungen, 50
- Netzwerkverbindung, 137
- Neues Gerät hinzufügen, 130
- Passwortschutz, 176
- PROFIBUS, 507
- PROFIBUS-Adresse, 507
- PROFIBUS-Port, 507
- PROFINET, 147
- Programmkarte, 122
- Richtlinien für Potentialtrennung, 68
- Schaltflächen RUN/STOP, 40
- Signalboard (SB), 25
- Signalmodul (SM), 25
- Startwerte eines DBs zurücksetzen, 728
- Thermischer Bereich, 46, 50
- TS-Adapter, 22
- Übersicht über die CPU, 19
- Übertragungskarte, 119
- Verdrahtungsrichtlinien, 67, 69
- Vergleichstabelle der CPU-Varianten, 20
- Verlorenes Passwort, 127
- Werte eines DBs erfassen, 728
- Zugriffsschutz, 176
- Zykluszeit, 89
- S7-Kommunikation
  - Verbindung konfigurieren, 139
- SB 1221
  - SB 1221 DI 4, 200 kHz, Schaltplan, 833
- SB 1222
  - SB 1222 DO 4 x 24 VDC, 200 kHz, Schaltplan, 836
- SB 1223
  - SB 1223 DI 2 / DO 2, Schaltplan, 842
  - SB 1223 DI 2 / DO, 200 kHz, Schaltplan, 839
- SB 1231 AI 1 x 16 Bit Thermoelement
  - Filterauswahltabelle, 853
- SB 1232
  - Schaltplan SB 1232 AO 1 x 12 Bit, 846
- SCALE\_X (Skalieren), 231
- Schachtelungstiefe, 73
- Schaltflächen RUN/STOP, 40
- Schaltpläne
  - CPU 1211C, 759
  - CPU 1212C, 769
  - CPU 1214C, 779
  - CPU 1215C, 791
  - SB 1221 DI 4, 200 kHz, 833
  - SB 1222 DO 4, 200 kHz, 836
  - SB 1223 DI 2 / DO 2, 842
  - SB 1223 DI 2 / DO 2, 200 kHz, 839
  - SB 1231 AI 1 x 16 Bit RTD, 856
  - SB 1231 AI 1 x 16 Bit Thermoelement, 853
  - SB 1231 AI x 12 Bit, 844
  - SB 1232 AO 1 x 12 Bit, 846
  - SM 1221 Signalmodul, 793
  - SM 1222-Signalmodul, 797
  - SM 1223 Signalmodul, 802, 805
  - SM 1231 AI 4 x 16 Bit TC, 822
  - SM 1231 AI 8 x 16 Bit TC, 822
  - SM 1231 Analogeingang, 812
  - SM 1231 RTD 4 x 16 Bit, 828
  - SM 1231 RTD 8 x 16 Bit, 828
  - SM 1232 Analogausgang, 813
  - SM 1234 Analogein-/Analogausgang, 816
- Schneller Zähler
  - Funktionsweise, 359
  - HSC, 357
  - Konfiguration, 366
- Schneller Zähler (HSC)
  - Kann nicht geforct werden, 734
- Schnittstellenkonfiguration
  - Anweisungen, 631
- Schutz
  - CPU, 176
  - Knowhow-Schutz für einen Codebaustein, 177
- Schutzart, 747

- Schutzstufe
  - Codebaustein, 177
  - CPU, 176
  - Mit CPU oder Memory Card verknüpfen, 178
  - Verlorenes Passwort, 127
- SCL (Structured Control Language)
  - ABS (Absolutwert bilden), 214
  - ACOS (Arcuscosinus oder inverser Cosinus), 216
  - Adressierung, 168
  - Algorithmus PID\_3Step, 368, 376
  - Algorithmus PID\_Compact, 368, 370
  - Arithmetik, 211
  - ASIN (Arcussinus oder inverser Sinus), 216
  - ATAN (Arcustangens oder inverser Tangens), 216
  - ATH (ASCII-Zeichenkette in Hexadezimalzahl wandeln), 280
  - ATTACH und DETACH, 305
  - Ausdrücke, 168
  - Bausteine aufrufen, 158
  - Bedingungen, 168
  - Betriebsstundenzähler, 265
  - Bitverknüpfung, 185
  - CAN\_DINT (Verzögerungsalarm stornieren), 312
  - CASE, 236
  - CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 230
  - CONCAT (Verketten), 283
  - CONTINUE, 240
  - CONV (Umwandeln), 225
  - COS (Cosinus), 216
  - CTD (Rückwärtszählen), 201
  - CTU (Vorwärtszählen), 201
  - CTUD (Vorwärts- und Rückwärtszählen), 201
  - DataLogClose (Datenprotokoll schließen), 340
  - DataLogCreate (Datenprotokoll erstellen), 335
  - DataLogNewFile (Datenprotokoll basierend auf vorhandenem Datenprotokoll erstellen), 342
  - DataLogOpen (Datenprotokoll öffnen), 338
  - DataLogWrite (Datenprotokoll schreiben), 341
  - DEC (Dekrementieren), 213
  - DECO (Decodieren), 254
  - DELETE (Teilzeichenkette löschen), 285
  - DeviceStates, 318
  - DIS\_AIRT (Alarmbearbeitung deaktivieren), 314
  - EN und ENO (Signalfluss), 174
  - EN\_AIRT (Alarmbearbeitung aktivieren), 314
  - ENCO (Encodieren), 254
  - EXIT, 241
  - EXP (Exponentialwert), 216
  - EXPT (Potenzieren), 216
  - FB oder FC aufrufen, 168
  - FILL\_BLK, 223
  - FIND (Teilzeichenkette finden), 288
  - FLOOR, 230
  - FOR, 237
  - FRAC (Nachkommastellen), 216
  - Funktionsweise der Zeiten, 197
  - Ganzzahl erzeugen, 229
  - GET\_DIAG, 321
  - Gleitpunktarithmetik, 216
  - GOTO, 241
  - HTA (Hexadezimalzahl in ASCII-Zeichenkette wandeln), 281
  - IF-THEN, 235
  - IN\_RANGE (innerhalb eines Bereichs), 208
  - INC (Inkrementieren), 213
  - INSERT (Teilzeichenkette einfügen), 286
  - INV (Invertieren), 253
  - JMP\_LIST, 243
  - LED-Zustand, 316
  - LEFT (Linke Teilzeichenkette), 284
  - LEN (Länge), 282
  - LIMIT, 216
  - LN (Natürlicher Logarithmus), 216
  - MAX (Maximum), 215
  - MC\_ChangeDynamic, 418
  - MC\_CommandTable, 415
  - MC\_Halt, 404
  - MC\_Home, 401
  - MC\_MoveAbsolute, 406
  - MC\_MoveJog, 413
  - MC\_MoveRelative, 408
  - MC\_MoveVelocity, 410
  - MC\_Power, 397
  - MC\_Reset, 400
  - MID (Mittlere Teilzeichenkette), 284
  - MIN (Minimum), 215
  - MOD (Modulo), 212
  - ModuleStates, 319
  - MUX (Multiplexen), 256
  - N\_TRIG, 192
  - NEG (Negation), 213
  - NORM\_X (Normalisieren), 231
  - OK, 208
  - Operatoren, 168
  - OR, 252
  - OUT\_RANGE (außerhalb eines Bereichs), 208
  - P\_TRIG, 192
  - PID\_3STEP, 376
  - PID\_Compact, 370
  - Priorität von Operatoren, 168
  - Programmieditor, 167
  - Programmsteuerung, 234
  - QRY\_CINT (Weckalarm abfragen), 310

- RD\_LOC\_T (Lokalzeit lesen), 263  
RD\_SYS\_T (Systemzeit lesen), 263  
REPEAT, 239  
REPLACE (Teilzeichenkette ersetzen), 287  
RETURN, 242  
RIGHT (Rechte Teilzeichenkette), 284  
ROL und ROR (Links rotieren und Rechts rotieren), 259  
S\_CONV (Umwandlungen Zahlenwert in Zeichenkette), 269  
S\_MOV (Zeichenkette verschieben), 269  
SCALE\_X (Skalieren), 231  
SEL (Selektieren), 255  
SET\_CINT (Weckalarm setzen), 308  
SET\_TIMEZONE (Zeitzone setzen), 267  
Setzen und Rücksetzen, 188  
SHL und SHR (Links schieben und Rechts schieben), 258  
SIN (Sinus), 216  
SQR (Quadrat), 216  
SQRT (Quadratwurzel), 216  
SRT\_DINT (Verzögerungsalarm starten), 312  
Steuerungsanweisungen, 168, 234, 235, 236, 237, 238, 239, 240, 241, 242  
STRG\_VAL (Zeichenkette in Zahlenwert umwandeln), 269  
Swap, 224  
SWITCH, 244  
T\_ADD (Zeiten addieren), 262  
T\_COMBINE (Zeiten verknüpfen), 263  
T\_CONV (Zeiten umwandeln), 261  
T\_DIFF (Zeitdifferenz), 262  
T\_SUB (Zeiten subtrahieren), 262  
TAN (Tangens), 216  
Übersicht, 166  
Übersicht über den PID-Regler, 367  
Übertragen, 218  
UFILL\_BLK (Bereich ununterbrechbar befüllen), 223  
Umwandlungsanweisungen, 226  
UND, 252  
VAL\_STRG (Zahlenwert in Zeichenkette wandeln), 269  
Var-Abschnitt, 167  
Vergleichen, 207  
WHILE, 238  
WR\_SYS\_T (Systemzeit schreiben), 263  
XOR (Exklusiv-ODER), 252  
Zahl runden, 229  
Zähler, 201  
Zeiten, 193  
SEL (Selektieren), 255  
SEND\_CFG (Sendekonfiguration), 604  
SEND\_PTP (Punkt-zu-Punkt-Daten senden), 610  
    Parameter LENGH und BUFFER, 612  
Serielle Kommunikation, 599  
Service und Support, 3  
SET\_CINT (Weckalarm setzen), 308  
SET\_TIMEZONE (Zeitzone setzen), 267  
Setzen, 188  
SGN\_GET (RS232-Signale abrufen), 616  
SGN\_SET (RS232-Signale setzen), 617  
SHL und SHR (Links schieben und Rechts schieben), 258  
Sicherheit  
    Kopierschutz, 178  
    Mit CPU oder Memory Card verknüpfen, 178  
    Verlorenes Passwort, 127  
    Zugriffsschutz, 176  
Siemens-Sicherheitszertifikat importieren, 593  
Siemens-Sicherheitszertifikat, Webseiten, 537, 593  
Signalboard (SB)  
    Ausbau, 55  
    Darstellung Analogausgang (Spannung), 819, 849  
    Darstellung Analogausgang (Strom), 819, 849  
    Darstellung Eingang (Spannung), 818, 848  
    Darstellung Eingang (Strom), 818, 848  
    Einbau, 55  
    Konfiguration von Parametern, 136  
    Leistungsbedarf, 879  
    Module hinzufügen, 132  
    SB 1221 DI 4, 200 kHz, 832  
    SB 1222 DO, 200 kHz, 834  
    SB 1223 DI 2 / DO 2, 840  
    SB 1223 DI 2 / DO 2, 200 kHz, 837  
    SB 1231 AI 1 x 12 Bit, 843  
    SB 1231 AI 1 x 16 Bit RTD, 854  
    SB 1231 AI 1 x 16 Bit Thermoelement, 850  
    SB 1232 AO 1x12 Bit, 845  
    Schrittantwortzeiten, 847  
    Übersicht, 25  
Signalmodul (SM)  
    Ausbau, 58  
    Darstellung Analogausgang (Spannung), 819, 849  
    Darstellung Analogausgang (Strom), 819, 849  
    Darstellung Analogeingang (Spannung), 818, 848  
    Darstellung Analogeingang (Strom), 818, 848  
    Einbau, 57  
    Erweiterungskabel, 61  
    Konfiguration von Parametern, 136  
    Leistungsbedarf, 879  
    Module hinzufügen, 132  
    Schrittantwortzeiten, 817  
    SM 1221, 792  
    SM 1222, 795, 797

- SM 1222 DO8 RLS, Umschaltung, 795
- SM 1223, 802
- SM 1231 AI 4 x 13 Bit, 808
- SM 1231 AI 4 x 16 Bit TC, 820
- SM 1231 AI 4 x RTD x 16 Bit, 826
- SM 1231 AI 8 x 16 Bit TC, 820
- SM 1231 AI 8 x RTD x 16 Bit, 826
- SM 1232 AO 2 x 14 Bit, 812
- SM 1232 AO 4 x 14 Bit, 812
- SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 814
- Technische Daten SM 1223, 805
- Übersicht, 25
- Signalverarbeitungsfehler, 617, 618
- Simulatoren, 876
- SIN (Sinus), 216
- SM 1231 RTD
  - Auswahltabellen, 831, 858
- SM und SB
  - Gerätekonfiguration, 129
  - Vergleichstabelle, 22
- SMS, 703
- Software-Flusskontrolle, 621
- Sonderzeichen
  - Benutzerdefinierte Webseiten, 565
- Spalten und Überschriften in Taskcards, 38
- Speicher
  - A (Prozessabbild der Ausgänge), 98
  - Adressen der Peripherieeingänge (Forcetabelle), 732
  - Arbeitsspeicher, 89
  - Beobachten der Speicherauslastung, 723
  - E (Prozessabbild der Eingänge), 98
  - L (Lokaldatenspeicher), 96
  - Ladespeicher, 89
  - M (Bitmerker), 99
  - Remanenter Speicher, 90
  - Systemspeicher, 92
  - Taktmerker, 92
  - Temporärer Speicher, 100
- Speicherbereich A
  - Impulsausgänge, 330
  - Impulskanäle konfigurieren, 332
- Speicherbereich E
  - Adressen der Peripherieeingänge (Forcetabelle), 732
  - Beobachten, 727
  - Beobachtungstabelle, 727
  - Forcefunktion, 733
  - Forcen, 732
  - Forcetabelle, 732
  - HSC (schneller Zähler), 359
  - KOP beobachten, 728
- Speicherbereiche
  - Adressierung von Booleschen Werten oder Bitwerten, 97
  - Direkter Zugriff, 96
  - Prozessabbild, 96
- Speicherkarte
  - Anlaufparameter konfigurieren, 119
  - Bestellnummer, 875
  - Technische Daten, 875
  - Verlorenes Passwort, 127
- Sprachen wechseln, benutzerdefinierte Webseiten, 583
- Sprachen, benutzerdefinierte Webseiten, 583
- Sprachumschaltung für benutzerdefinierte Webseiten programmieren, 584
- Sprungmarke, 242
- SQR (Quadrat), 216
- SQRT (Quadratwurzel), 216
- SRT\_DINT (Verzögerungsalarm starten), 312
- Standard-Hutschiene, 51
- Standard-Webseite "CPU-Identifikation",
- Standard-Webseite "Datenprotokolle",
- Standard-Webseite "Diagnose",
- Standard-Webseite "Einführung",
- Standard-Webseite "Identifikation",
- Standard-Webseite "Kommunikation",
- Standard-Webseite "Modulinformationen",
- Standard-Webseite "Start",
- Standard-Webseite "Variablenzustand",
- Standard-Webseiten, 531
  - An- und Abmelden, 536
  - Aufbau, 535
  - Betriebszustand wechseln, 538
  - Cookie-Einschränkungen, 593
  - Datenprotokolle, 546
  - Diagnose, 540
  - Einführung, 537
  - Identifikation, 539
  - JavaScript-Einschränkungen, 592
  - Kommunikation, 543
  - Modulinformationen, 540
  - Sicherer Zugriff, 534
  - Start, 538
  - Variablenzustand, 544
  - Zugriff über PC, 533
- Startbedingungen, 624
- STARTUP, Betriebszustand
  - Forcefunktion, 733
- Startwerte eines DBs zurücksetzen, 728
- Steckverbinder
  - Einbau und Ausbau, 60
- Steigende Flanke, 191
- STEP 7

- Ändern von Einstellungen, 39
  - Anfangswert eines FB, 161
  - Anlaufverarbeitung, 78
  - Anzahl der Codebausteine, 73
  - ASi,
  - AS-i-Port, 511
  - Bausteinaufrufe, 73
  - Bausteine aus einer Online-CPU kopieren, 179
  - Bedienpanel, 40
  - Beobachten, 727, 728
  - Betriebsstundenzähler, 265
  - Betriebszustände, 75
  - Codebausteine im Anwenderprogramm aufrufen, 158
  - CPU konfigurieren, 133
  - Datenbaustein (DB), 73
  - Drag & Drop zwischen Editoren, 39
  - Eigenschaft Uhrzeitsynchronisation (PROFINET), 153
  - Einfügen von Anweisungen, 33
  - Ergänzen von Eingängen oder Ausgängen in einer KOP- oder FUP-Anweisung, 37
  - Erweiterbare Eingänge und Ausgänge, 37
  - Ethernet-Port, 147
  - Favoriten, 34
  - Forcefunktion, 733
  - Forcen, 732
  - Funktion (FC), 160
  - Funktionsbaustein (FB), 73, 161
  - Funktionsweise, 729
  - Gerätekonfiguration, 129
  - Gültige FC-, FB- und DB-Nummern, 73
  - HSC-Konfiguration, 366
  - Instanz-Datenbaustein (DB), 161
  - Kommunikationslast, 89
  - Laden in CPU, 179
  - Lineare und strukturierte Programme, 156
  - Memory Card, 117, 875
  - Module hinzufügen, 132
  - Module konfigurieren, 136
  - Netzwerkverbindung, 137
  - Neues Gerät hinzufügen, 130
  - Nicht gesteckte Module, 43
  - Passwortschutz, 177
  - Portalansicht, 32
  - Prioritätsklasse (OB), 80
  - PROFIBUS, 507
  - PROFIBUS-Port, 507
  - PROFINET, 147
  - PROFINET IO-Gerät hinzufügen, 495
  - Programmkarte, 117
  - Projektansicht, 32
  - RD\_LOC\_T (Lokalzeit lesen), 263
  - RD\_SYS\_T (Systemzeit lesen), 263
  - Schaltflächen RUN/STOP, 40
  - Startwerte eines DBs zurücksetzen, 728
  - Übertragungskarte, 117
  - Vergleichen und synchronisieren, 725
  - Werte eines DBs erfassen, 728
  - WR\_SYS\_T (Systemzeit schreiben), 263
  - Zähler, 201
  - Zuweisen einer IP-Adresse zu einer Online-CPU, 145
  - Zykluszeit, 89
  - Zykluszeit, 89
  - STEP 7 Webseiten, 4
  - STEP 7-Programmierung
    - Benutzerdefinierte Webseiten, 569
    - PtP-Beispielprogramm, 640
  - Steuer-DB für benutzerdefinierte Webseiten
    - Anforderungsbefehle und -zustände, 587
    - Globale Befehle, 587
    - Parameter WWW-Anweisung, 569
  - Stoppbits, 620
  - STP (Zyklus stoppen), 248
  - STRG\_VAL (Zeichenkette in Zahlenwert umwandeln), 269
  - String
    - Datentyp String, 268
    - Datentyp STRING, 107
    - S\_MOVE (Zeichenkette verschieben), 269
    - Übersicht über Zeichenkettenoperationen, 282
  - Strukturierte Programmierung, 156, 158
    - Bausteine aufrufen, 158
  - SUB (Subtrahieren), 211
  - Subnetzmaske, 147
  - Support, 3
  - SWAP, 224
  - SWITCH, 244
  - Synchronisation
    - Eigenschaft Uhrzeitsynchronisation (PROFINET), 153
  - Systemspeicherbyte, 93
  - Systemuhr
    - RD\_LOC\_T (Lokalzeit lesen), 263
    - RD\_SYS\_T (Systemzeit lesen), 263
    - WR\_SYS\_T (Systemzeit schreiben), 263
  - Systemvoraussetzungen, 32
- T**
- T\_ADD (Zeiten addieren), 262
  - T\_COMBINE (Zeiten verknüpfen), 263
  - T\_CONFIG, 479

- T\_CONV (Zeiten umwandeln), 261
- T\_DIFF (Zeitdifferenz), 262
- T\_SUB (Zeiten subtrahieren), 262
- Tageszeit
  - CPU online konfigurieren, 721
- Takt
  - Merkerbyte, 94
- TAN (Tangens), 216
- Taskcards
  - Spalten und Überschriften, 38, 678
- TCON, 466
  - Konfiguration, 138
  - Verbindungs-IDs, 454
  - Verbindungsparameter, 140
- TCON\_Param, 140
- TCP
  - Ad-hoc-Modus, 458
  - Parameter, 140
  - Protokolle, 457
  - Verbindungs-IDs, 454
  - Verbindungskonfiguration, 138
- TCP/IP-Kommunikation, 457
- TDISCON, 466
- Technische Daten, 743
  - Allgemeine technische Daten, 743
  - ATEX-Zulassung, 744
  - Bemessungsspannungen, 748
  - CB 1241 RS485, 870
  - CE-Zulassung, 743
  - CM 1241 RS232, 873
  - CPU 1211C, 750
  - CPU 1212C, 760
  - CPU 1214C, 770
  - CPU 1215C, 780
  - C-Tick-Zulassung, 745
  - cULus-Zulassung, 744
  - Darstellung Analogausgang (Spannung), 819, 849
  - Darstellung Analogausgang (Strom), 819, 849
  - Darstellung Analogeingang (Spannung), 818, 848
  - Darstellung Analogeingang (Strom), 818, 848
  - Eingangssimulatoren, 876
  - Elektromagnetische Verträglichkeit (EMV), 746
  - FM-Zertifizierung, 744
  - Industrienumgebungen, 745
  - Lebensdauer eines Relais, 749
  - Memory Cards, 875
  - SB 1221 DI 4, 200 kHz, 832
  - SB 1222 DO 4, 200 kHz, 834
  - SB 1223 DI 2 / DO 2, 840
  - SB 1223 DI 2 x / DO 2, 200 kHz, 837
  - SB 1231 AI 1 x 12 Bit, 843
  - SB 1231 AI 1 x 16 Bit RTD, 854
  - SB 1231 AI 1 x 16 Bit Thermoelement, 850
  - SB 1232 AO 1x12 Bit, 845
  - Schaltplan SB 1231 AI 1 x 12 Bit, 844
  - Schaltplan SB 1231 AI 1 x 16 Bit RTD, 856
  - Schaltplan SB 1231 AI 1 x 16 Bit Thermoelement, 853
  - Schaltplan SM 1231 AI 4 x 16 Bit TC, 822
  - Schaltplan SM 1231 AI 8 x 16 Bit TC, 822
  - Schaltplan SM 1231 RTD 4 x 16 Bit, 828
  - Schaltplan SM 1231 RTD 8 x 16 Bit, 828
  - Schaltpläne SM 1231 Analogeingang, 812
  - Schaltpläne SM 1232 Analogausgang, 813
  - Schaltpläne SM 1234 Analogein-/Analogausgang, 816
  - Schrittantwortzeiten (CPU), 756, 766, 776, 786
  - Schrittantwortzeiten (SB), 847
  - Schrittantwortzeiten (SM), 817
  - Schutz, 747
  - Signalmodul SM 1231 AI 4 x 16 Bit TC, 820
  - Signalmodul SM 1231 AI 4 x RTD x 16 Bit, 826
  - Signalmodul SM 1231 AI 8 x 16 Bit TC, 820
  - Signalmodul SM 1231 AI 8 x RTD x 16 Bit, 826
  - SM 1221 Signalmodul, 792
  - SM 1221, Schaltplan, 793
  - SM 1222 DO8 RLS, Umschaltung, 795
  - SM 1222 Signalmodul, 795, 797
  - SM 1222, Schaltplan, 797
  - SM 1223 Signalmodul, 802, 805
  - SM 1223, Schaltplan, 802, 805
  - SM 1231 AI 4 x 13 Bit, 808
  - SM 1232 AO 2 x 14 Bit, 812
  - SM 1232 AO 4 x 14 Bit, 812
  - SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 814
  - Umgebungsbedingungen, 746
  - Zulassung für das Seewesen, 745
- Technische Daten des CM 1241 RS485, 870
- Technische Daten für das digitale Signalboard (SB)
  - SB 1221 DI 4, 200 kHz, 832
  - SB 1222 DO 4, 200 kHz, 834
  - SB 1223 DI 2 / DO 2, 840
  - SB 1223 DI 2 / DO 2, 200 kHz, 837
- Technischer Support, 3
- Technischer Support von Siemens, 3
- Technologische Objekte
  - HSC (schneller Zähler), 359
- Telecontrol, 700
- TeleService über GPRS, 700
- Teleservice-Kommunikation
  - TM\_MAIL, 709
- Terminalemulator für PtP-Beispielprogramm, 641
- Testen
  - Laden im Betriebszustand RUN, 735, 741



- Thermischer Bereich, 46, 50
- Thermoelement
- Filterauswahltabelle SB 1231 Thermoelement, 852
  - Filterauswahltabelle SM 1231 Thermoelement, 824
  - Grundlegende Funktionsweise, 824, 852
  - Kompensation der kalten Verbindungsstelle, 824, 852
  - SB 1231 Thermoelement-Auswahltabelle, 852
  - SM 1231 Thermoelement-Auswahltabelle, 824
- TIA-Portal
- Portalansicht, 32
  - Projektansicht, 32
- Time
- Datentyp Time, 105
  - DTL (Datentyp Date and Time long), 106
  - RD\_LOC\_T (Lokalzeit lesen), 263
  - RD\_SYS\_T (Systemzeit lesen), 263
  - SET\_TIMEZONE (Zeitzone setzen), 267
  - T\_ADD (Zeiten addieren), 262
  - T\_DIFF (Zeitdifferenz), 262
  - T\_SUB (Zeiten subtrahieren), 262
  - TOD (Datentyp Time of Day), 105
- TM\_MAIL, 709
- TRCV, 466
- Ad-hoc-Modus, 458
  - Verbindungs-IDs, 454
- TRCV\_C, 459, 494
- Ad-hoc-Modus, 458
  - Konfiguration, 138
  - Verbindungs-IDs, 454
  - Verbindungsparameter, 140
- TRUNC (Ganzzahl erzeugen), 229
- TS-Adapter, 22
- Einbau auf DIN-Schiene, 66
  - Einbau eines TS-Moduls, 63
  - SIM-Karte, 64
  - Wandmontage, 66
- TSAP, 457
- TSAP (Transport Service Access Point), 140, 459, 492, 522
- TSAPs
- eingeschränkt, 487
- TSEND, 466
- Verbindungs-IDs, 454
- TSEND\_C, 493
- Konfiguration, 138
  - Verbindungs-IDs, 454
  - Verbindungsparameter, 140
- TSEND\_C, 493
- TURCV, 474
- Konfiguration, 138
  - Verbindungsparameter, 140
- TUSEND, 474
- Konfiguration, 138
  - Parameter, 140
- ## U
- Übertragungsbaustein (T-Baustein), 493
- Übertragungskarte, 119
- Anlaufparameter konfigurieren, 119
  - Bestellnummer, 875
  - Funktionsweise, 117
  - In CPU einfügen, 118
  - Leere Übertragungskarte für ein verlorenes Passwort, 127
  - Übersicht, 117
  - Verlorenes Passwort, 127
- Überwachung
- Anweisung LED, 316
- UDP
- Parameter, 140
  - Verbindungskonfiguration, 138
- UDP-Protokoll, 457
- UFILL\_BLK (Bereich ununterbrechbar befüllen), 223
- Uhr
- Echtzeituhr, 94
  - RD\_LOC\_T (Lokalzeit lesen), 263
  - RD\_SYS\_T (Systemzeit lesen), 263
  - WR\_SYS\_T (Systemzeit schreiben), 263
- Uhrzeit
- T\_COMBINE (Zeiten verknüpfen), 263
- Umbruch, 623
- Umgebungsbedingungen
- Betriebsbedingungen, 746
  - Industrienumgebungen, 745
  - Transport und Lagerung, 746
- Umwandlung (SCL-Anweisungen), 226
- UND, 252
- ## V
- VAL\_STRG (Zahlenwert in Zeichenkette wandeln), 269
- Variablen
- Auf dem PC überwachen und ändern, 544
  - Beobachten, 727
  - Forcefunktion, 733
  - Forcen, 732
  - Zustand, 727
- Variablen auf dem PC ändern, 544
- Variablen auf dem PC überwachen, 544
- Variablenindex für ein Array, 222
- Verbindung zwischen lokaler und Partner-CPU, 451

Verbindungen mit mehreren Teilnehmern  
 Ethernet-Protokolle, 522  
 Verbindungsarten, 522  
 Verbindungen, Webserver, 591  
 Verdrahtungsrichtlinien, 69  
 Abstand für Luftströmung und Kühlung, 46  
 Erdung, 69  
 Voraussetzungen, 67  
 Vergleichen, 207  
 Vergleichstabelle  
 CPU-Varianten, 20  
 HMI-Geräte, 28  
 Module, 22  
 Verlorenes Passwort, 127  
 Verschiedene PtP-Parameterfehler, 601  
 Versionen von Anweisungen, 38, 678  
 Verzeichnisse, Sprachen für benutzerdefinierte  
 Webseiten, 584  
 Verzögerungsalarm, 312  
 Visualisierung  
 HMI-Geräte, 28  
 Voraussetzungen für die Installation, 32  
 Voraussetzungen PC, 32  
 Voraussetzungen, Installation, 32

**W**

Warmstart, 75  
 Warteschlangen, 81  
 Wartezeit, 620  
 Watchdog, 247  
 Webseiten  
 STEP 7, 4  
 Webserver, 531  
 Aktivieren, 532  
 Aktualisierungsrate, 532  
 Einschränkungen, 591  
 Konventionen bei Anführungszeichen, 565  
 maximale Anzahl HTTP-Verbindungen, 591  
 Standard-Webseiten, 533  
 Webserver, benutzerdefinierte Webseiten, 551  
 Wechsel von RUN nach STOP, 95  
 Weckalarm-OB, 80  
 Werte eines Online-DBs erfassen, 728  
 Werte in der Beobachtungstabelle auslösen, 731  
 WR\_SYS\_T (Systemzeit schreiben), 263  
 WRIT\_DBL, 353  
 WRREC, 290, 296  
 WWW (benutzerdefinierte Webseiten aktivieren), 569

**X**

XON / XOFF, 621  
 XOR (exklusives ODER), 252

**Z**

Zahlen  
 Binär, 103  
 Ganzzahl, 104  
 Real, 104  
 Zähler  
 Anzahl, 21, 752, 762, 772, 782  
 Funktionsweise des HSC, 359  
 Größe, 22, 752, 762, 772, 782  
 HSC (schneller Zähler), 357  
 HSC-Konfiguration, 366  
 Zeichen für das Ende einer Meldung, 628  
 Zeichen für den Beginn einer Meldung, 624  
 Zeichenabstand, 628  
 Zeichenfolge  
 Meldungsbeginn, 624  
 Meldungsende, 628  
 Zeichenposition  
 Meldungslänge, 629  
 Zeit  
 T\_CONV (Zeiten umwandeln), 261  
 WR\_SYS\_T (Systemzeit schreiben), 263  
 Zeit akkumulieren (TONR), 193  
 Funktionsweise, 197  
 Zeit zurücksetzen (RT), 193  
 Zeiten  
 Anzahl, 21, 752, 762, 772, 782  
 Funktionsweise, 197  
 Größe, 21, 752, 762, 772, 782  
 RT (Zeit zurücksetzen), 193  
 TOF (Ausschaltverzögerung), 193  
 TON (Einschaltverzögerung), 193  
 TONR (Zeit akkumulieren), 193  
 TP (Impuls), 193  
 Zeitfehler-OB, 81  
 Zugreifen  
 Benutzerdefinierte Webseiten, 571  
 Datenprotokolle auf PC, 546  
 Zugriffsschutz  
 CPU, 176  
 Zulassung für das Seewesen, 745  
 Zulassungen  
 ATEX-Zulassung, 744  
 CE-Zulassung, 743  
 C-Tick-Zulassung, 745  
 cULus-Zulassung, 744

- FM-Zertifizierung, 744
- Zulassung für das Seewesen, 745
- Zurücksetzen auf Werkseinstellungen, 722
- Zustand
  - Anweisung LED, 316
  - LED-Anzeigen, 717
  - LED-Anzeigen (Kommunikationsschnittstelle), 597
- Zyklus
  - Forcefunktion, 733
  - Forcen, 732
- Zykluszeit
  - Beobachten, 723
  - Konfiguration, 89
  - Übersicht, 87

